GB99/01772

The Patent Office

INVESTOR IN PEOPLE

5

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 25th June, 1999

Patents Form 1/77

The Patent Office

1/77

Patents Act 1977
(Rule 16)

The Patent Office
Cardiff Road
Newport
Gwent NP9 1RH

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

1. Your reference

A25753

2. Patent application number
(The Patent Office will fill in this part)

9902648.6 - 5 FEB 1999

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

BRITISH TELECOMMUNICATIONS public limited c mpany
81 NEWGATE STREET
LONDON, EC1A 7AJ, England
Registered in England: 1800000

Patents ADP number *(if you know it)*

1867002

If the applicant is a corporate body, give the country/state of its incorporation

UNITED KINGDOM

4. Title of the invention

COMMUNICATIONS NETWORK

5. Name of your agent *(if you have one)*

LIDBETTER, TIMOTHY GUY EDWIN

"Address for Service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

BT GROUP LEGAL SERVICES
INTELLECTUAL PROPERTY DEPARTMENT
HOLBORN CENTRE
120 HOLBORN
LONDON, EC1N 2TE

Patents ADP number *(if you know it)*

1867001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day/month/year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
(See note (d))

YES

9. Enter the number of sheets for any following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form		
Description	**48**	
Claim(s)	**6**	✓ 8H
Abstract	**1**	
Drawing(s)	**15**	+15

10. If you are also filing any of the following, state how may against each item

Priority Documents

Translations of priority documents

Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*

Request for preliminary examination and search *(Patents Form 9/77)*

Request for substantive examination *(Patents Form 10/77)*

Any other documents *(please specify)*

11. I/We request the grant of a patent on the basis of this application.

Signature(s) Date:

05 February 1999

Timothy Guy Edwin LIDBETTER, Authorised Signat ry

12. Name and daytime telephone number of person to contact in the United Kingdom **Bhavna Vasani** **0171 492 8147**

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

COMMUNICATIONS NETWORK

The present invention relates to a communications network, and in particular to charging mechanisms in such a network. It includes aspects of the inventions disclosed and claimed in the present applicant's co-pending British patent application no. 9812161.9 filed 5 June 1998 and the contents of that earlier application are incorporated herein by reference.

In conventional communications networks, such as national PSTNs (public switched telephone networks), a significant proportion of the network resources are devoted to metering and billing network usage. Studies have estimated these resources as consuming as much as 6% of the revenue of a telecommunications company. The Internet, by contrast, does not in general incorporate metering and billing mechanisms for individual customers. The absence of the network infrastructure required to support metering and billing reduces the operational costs of the Internet compared to conventional telephony networks, and has facilitated the rapid expansion of the Internet . However the absence of appropriate billing mechanisms has significant disadvantages in terms of the characteristics of the traffic carried by the internet.It encourages profligate use of network resources, and diminishes the incentive for investment in network infrastructure to support new applications requiring, e.g., guaranteed quality of service (QoS) and led to subsription based Internet access services.

According to a first aspect of the present invention, there is provided a method of operating a communications network comprising:

a) measuring at each of a plurality of customer terminals usage by the the respective customer terminal of network resources; and

b) subsequently calculating a network usage charge from the measurement data generated by step (a).

The present inventors have found that a key step in implementing a lightweight charging protocol suitable for use in a federated network is to de-centralise the metering of network usage by arranging for each customer terminal to monitor its own use of network resources. In this way a charging mechanism is provided that is intrinsically scaleable and that avoids significant overheads within the network.

Moreover, the invention, in preferred implementations, provides a basis for a multi-service network packet network in which it is not necessary to police every packet. This makes it far easier to implement a multi-service network, i.e. one in which different packets may be scheduled differently according to which class of service applies, than with existing schemes.

Preferably the method includes storing the measurement data generated by step (a). Preferably there is stored with the measurement data data identifying a tariff applicable to the said measurement data. The said data identifying the tariff may be the tariff itself, or may take the form of some identifying code or pointer for the tariff. Storing the tariff enables accounting data to be generated from measurements at the customer terminal even if the tariff varies over time.

Preferably the method includes communicating data generated by step (a) to a network accounting object controlled by a network operator. Alternatively data may be communicated from the network operator to the customer in a conventional way. The network usage data may be communicated explicitly and the charge for network usage calculated by the network operator. Alternatively the usage data may be communicated implicitly in accounting data indicating a charge calculated by the customer terminal.

Preferably the method includes a step carried out by the network operator of sampling part only of the traffic communicated between a customer terminal and the network. This sampled traffic is then compared with the network usage data reported from the customer terminal to the network provider accounting object, thereby detecting any discrepancy. The comparison may be of the total charged for network usage, or may be of the detailed measurement data. The former may be the norm for efficiency, with the latter used, in this case, only if the former shows discrepancies, in order to store evidence of fraud.

The inventors have found that the efficiency of the charging process can be further enhanced if the customer is responsible for measuring usage and providing useage data or priced useage data and the network operator measures only a sample of the customer traffic, on a random basis, to confirm the reliability of data provided by the customer.

Preferably the network operator accounting object is configurable to receive data either from a measurement object controlled by the network operator or from a customer terminal. Preferably the method includes changing from one configuration to the other in response to a control signal received at the network accounting object.

Preferably the method includes communicating measurement data to a system remote from the customer terminal. For example, data may be communicated from a number of customer terminals to a corporate accounting system. The data may be sent explicitly, and/or a usage charge calculated using the data may be sent to the remote system. When data is reported to a remote system, this may be done immediately the data is generated, or may be done in the form of a report aggregating data from a series of measurements over a period of time.

Preferably the method includes:

communicating traffic between a customer terminal and a first network domain connected to the customer terminal,

further communicating the said traffic between the first network domain and a second network domain connected to the first network domain;

communicating network usage data from the customer terminal to a first network accounting object in the first domain;

communicating accounting data between the first network accounting object and a second network accounting object in the second domain.

This aspect provides a powerful and efficient method of accounting between domains in a federated data network. Although data may be flowing e.g from a first customer terminal, via intermediate network domains to a second customer terminal, the accounting data (i.e. the measurement data or data derived therefrom) need not all flow in the same direction. The invention encompasses, for example, systems in which accounting data is passed from the customer to the first domain and also is passed from the second network domain to the first network domain.

Preferably the method includes determining from a current routing table in the first network domain the identity of a second domain communicating data with the customer terminal via the first network domain, and communicating accounting data for the customer terminal with the second domain identified by the current routing table.

According to another aspect of the present invention, there is provided a method of operating a network comprising a plurality of network domains, including calculating a charge for use by a respective customer of network resources, and making payment in settlement of the said charge to a third party clearer. This clearing payment may be used to apportion charges between the end users in any desired ratio, e.g. the sender pays all, or sender pays 60 % receiver pays 40%, etc. .

According to a further aspect of the present invention, there is provided a method of operating a packet network providing a plurality of different service levels, the method including including passing the said packets through a packet router, and in the packet router determining a classification of packets, scheduling packets differently depending on the packet classification and, at a location remote from the router, policing the service levels of packets to determine the eligibility of a packet for a respective service class .

The invention also encompasses communications networks arranged to operate by the methods of the invention, and customer terminals, and network accounting servers , and routers for use in such a network.

Systems embodying the present invention will now be described in further detail, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a schematic showing a network embodying the invention;

Figures 2a and 2b are schematics showing the component objects of a charging architecture for use with the network of Figure 1;

Figure 3a and 3b show data passed between the accounting objects of Figure 2a;

Figure 4 is a schematic showing protocol stacks on a customer terminal and in the network domain;

Figures 5a to 5e are class diagrams for software implementing accounting and measurement objects;

Figure 6 is a diagram showing a graphic user interface (GUI) for use with the objects of Figures 5a to 5e;

Figure 7 is a diagram showing the interface between neighbouring domains of the network of Figure 1;

Figure 8 is a diagram showing schematically the distribution of accounting data through multiple network domains;

Figure 9 is a diagram showing a network using service provider clearing;

Figure 10 is a diagram showing a network using third party clearing

As shown in Figure 1, a communications network 1 includes a number of network sub-domains 2A-C. The network sub-domains may be under the control of different operators. The operation of the network does not assume that there is mutual trust between the different operators. The network subdomains are interconnected by gateway routers 3, 4. In the present example the communications network is the Internet and supports both unicast and multicast Internet Protocol (IP) and associated protocols. A customer terminal 5 is connected via a public switched telephony network (PSTN) 6 and an access router 7 to a subdomain 2A. No policing is required at the access router. The gateway routers 3,4, and access router 7 may be commercially available devices such as CISCO series 7500 routers and CISCO series AS5800 universal access server respectively. Other customer terminals are connected to the network, including a Java-enabled mobile terminal 8 and a data server 9.

The customer terminal 5 may be connected via a LAN to an accounting server. The accounting server may include an accounting object as described below that receives measurement data from the customer terminal.

Tariffs for the use of network resources are multicast through the network to the customer terminals. These tariffs are divided into bands of different volatilities. The tariffs are varied under the control of the network operators to reflect the overall loading of the network. That is to say, if network loading becomes high then the tariffs may be increased to reflect the scarcity of network resources. A network management platform 10 is connected to each subdomain. Each network management platform may comprise, for example, a computing system comprising a SPARC workstation running UNIX (Solaris) together with network management applications. The network management platform 10 hosts management entities and tariff entities. It may also function as an accounting server hosting network accounting objects as described below. The network management platform communicates with agents 100 in managed devices connected to the respective subdomain, for example using SNMP (simple network management protocol). The management platform monitors the overall loading of network resources in the respective subdomains, and adjusts the tariffs for network use accordingly. The Net management platform (NMP) instructs the agent to monitor the device and report aggregated results at regular intervals back to the NMP, so the NMP can monitor the combination of all reports.

In addition to this central control of the tariffs, a tariff algorithm at each customer terminal may be arranged to respond automatically to a locally detected variation in the loading of network resources. The use of local tariff variation is described and claimed in the present Applicant's co-pending application also entitled "Communications Network", BT reference A25626.

In the present example, charging is carried out using a "pay and display" process but traditional payment methods can alternatively be used. .. Figures 2a and 2b show the objects used to implement the charging architecture in this case. Figure 2a shows the higher level objects and 2b shows the component objects used in a software implementation of the architecture of Figure 2a and expands further the distribution of the accounting objects within a single domain. In Figure 2a, objects on the client terminal are shown in the half of the Figure labelled "customer" and objects on the access router 7 and the corresponding network sub-domain are shown in the half of the Figure labelled "edge network". The objects on the customer terminal include a session control object S, a customer business rules object B_c , a customer pricing object Pr_c, a QoS manager Q, a customer accounting object Act_c and a customer measurement object M_c. The business rules object B_c receives information on those aspects of the session which involve liability for payment and receives current pricing data from the pricing object Pr_c. The customer business object makes decisions, under the customer's policy control on which chargeable services are utilised, and how much of the chargeable services are utilised. These decisions are fed to the QoS manager Q, which decides which mechanisms are used to achieve the requirements. The QoS manager (and the accounting object) then controls the customer measurement object M_c to determine which aspects of traffic and service to measure and which aspects to ignore. The measurement object then records the selected aspects of the traffic, for example counting the number of packets transmitted and received by the customer terminal and the QoS levels for those packets. These data, together with the current tariffs, including any premium for congestion, are then used by the customer terminal to determine the charge payable to the network operator. The measurement object M_c is also programmed , by the accounting object, with instructions that determine the frequency at which data is reported to the customer accounting object Act_c . The customer accounting object Act_c passes accounting information (priced or not) to an accounting object Act_p in the network provider's domain. On the network provider's side, that is to say within the subdomain to which the customer terminal is connected, the customer's traffic is measured by a version of M, denoted M_p, but only on a sampling basis determined by the policing function, Po. That is to say, the network operator samples the customer's traffic only intermittently. Po controls where in the network measurements are made in order to capture all of any particular customer's traffic. A bulk measurement function, M_b, is responsible for reporting aggregate traffic levels, as reflected in the moving average of the router queue lengths, to the pricing object, Pr_p. Bulk measurements would typically be collected from across the provider's domain to a centralised pricing function (which would be replicated for reliability). Pr_p sets prices taking into account the business rules from the network provider's business object, B_p, as well as the current traffic levels reported by Mb and pricing from neighbouring providers. The policing function, Po, compares sample measurements from M_p with accounting messages received at Act_p as a result of the customers own measurements . If it establishes that the accounts are insufficient it might restrict service at the access control gateway, Acs, or initiate some other punishment. Encapsulated within the accounting object another policing object checks the accounts match the payments within the contracted time for payment. Finally, the identity mapping function, I, provides a mapping between a customer's identity (account, digital signature, etc.) and their current network address (typically allocated by the ISP, whether unicast or multicast).

The measurement (M) objects provide to the accounting (Act) objects the information that is required to create firstly accounting records and subsequently reports and bills. Measurement records are not stored as such in the Act objects: measurement data is translated into accounting records as soon as possible. The translation of measurement data into accounting records involves a change of class type and some aggregation. In addition the measurement data may be linked to tariff information. The measurement data returned by the measurement objects includes, in this example, the following elements:

<u>IP addresses</u> of the two endpoints involved in the communication. This is readily available from the network packets.

Port numbers: These are used to distinguish between different services used by a user at one time. The port numbers are also available from the network packets.

Type of packets: service identity. This identifies the type of service, e.g. as RSVP, as differential service or as data. This information allows different tariffs to be applied depending on the packet type.

Network usage information. This is the measurement data itself and may comprise, for example, a count of the number of packets.

Time period information. This, if element, when used, indicates the length of time over which the measurement was made

Time reference. This may include a start time and an end time and may be used, for example, for applying discounts to traffic during defined "off-peak" hours.

In the presently preferred implementation, measurement data is returned by the measurement object to the Act object on an event-driven basis at time intervals controlled by the accounting object. Alternative approaches may use polling of the measurement object by the Act object, or event driven polling,: Communication of data may be effected using Java - RMI (remote method invocation) and the Java event model or a socket may be created between Act and M to send measurement objects . Further alternative communication mechanisms include the use of CORBA or SNMP like messaging. The present example makes use of an RMI/CORBA-like distributed event programming infrastructure called FLEXINET.

Measurement objects (M) offer a control interface to Act objects, so that Act objects can control what measures, and when and where M reports its measurement information.This control interface offers access to the following parameters:

1. Frequency at which measurement records are required (for a given customer or set of customers). ,This makes it possible to accommodate different accounting business models including, e.g., pay-as-you-go and traditional billing. The frequency may be specified as a period of a number of milliseconds.

2. What is to be reported to Act (for a given customer or set of customers). This parameter might specify all packets, or only packets with a give QoS threshold etc.

3. Where to report measurements (for a given customer or set of customers). This parameter may be a simple reference to the Act object or another business-related object for auditing or marketing purposes.

4. Current metering properties of the measurement object.

The Meter M at the network provider multiplexes the different measurement request for different customers and optimise the measurement and reporting processes.

The accounting objects on the customer terminal may be implemented using a small encrypted flat-file database. On the network provider's side, the equivalent objects may be implemented using a larger database that is scaleable to handle e.g., tens of thousands of customer accounts. An object request broker (ORB) is used for communication between the customer-side objects and the network-side objects, implemented using commercially available tools such as ORBIX (TradeMark) from Iona Technologies plc. Serialisation is used to stream objects from one database to another via the network. The process of serialisation takes all the attributes of an object and streams the attributes over a specified medium together with information specifying the type of object that originated the data. A process of de-serialisation then takes the data from the transmission medium together with the object type information and creates a new object of the specified type and fills it with the data. The accounting databases hold a set of serialised accounting objects. The larger database required by the network provider may be an object-oriented database that accepts objects and serialises them into its storage space. Alternatively a non object oriented database may be used, in which case the accounting objects are translated into database types. For example the accounting objects are translated into SQL data types for use with a relational database.

The serialisation/de-serialisation mechanism described above is also used to support the measurement and accounting interface between network domains. For example, the edge-of-network domain that communicates packets to and from the customer terminal in turn passes packets to a number of neighbouring domains. Just as accounting data is passed from the customer to the edge-of-network domain, so also accounting data is passed from an accounting object 71 in the edge-of-network domain to an accounting object 72 in a neighbouring domain, and payment is made by the operator of the edge-of-network domain to the operator of the neighbouring domain. In this context, the edge-of-network domain is a retail domain, and the neighbouring domains are wholesale domains. As shown in Figure 7, the architecture of the interface between the retail domain and the wholesale domains is a recursive version of the interface between the retail domain and the end customer. However all the measurement and QoS features of the interface to the end customer are not required in the interface between the retail and wholesale networks. Where, as in this example, there are multiple wholesale providers, then the current routing and/or address allocation in the retail network is interrogated to apportion accounting between the wholesale networks. This is effectively another form of identity mapping, I. The mapping is needed between the identities of each neighbour provider and their current groups of unicast addresses, address prefixes, multicast addresses or autonomous system (AS) numbers. This is not generally required in the edge architecture, as an edge customer typically has only one provider. If multiple providers were used by the customer, then mapping to apportion accounting is used at the edge too. As before, the measurement of traffic between retail and wholesale domains can be sampled and done in parallel to the data flow - no blocking is required. Any pair of network providers might in practice each be mutual customers. In this case, the architecture for the retail/wholesale interface is mirrored so that all functions operate in both directions. Any payments between network domains are then determined by the balance of the products of each accounting flow and the relevant prices.

In a network comprising multiple domains then, as shown in Figure 8, a "wholesale" domain 82 may receive accounting data from a number of retail networks 81,83.. These data are aggregated by the accounting object in domain 82 and then apportioned between further neighbouring domains, such as domain 84. The way in which the accounting data are apportioned is determined by an averaged border routing table maintained in the domain 82 Figures 3a and 3b show the data which are passed between the accounting objects. In this example the account data comprises: account identity; bill record identity; service type identifier; source address; destination address; tariff identity; time; period (i.e. the period covered by the bill record); units; cost; and currency. In addition, the payment data comprises the amount of money and the currency of payment.

Figure 4 shows the measurement region within protocol stacks on the customer terminal and in the retail network domain. Ideally there would be two measurement points within this region, one trusted by the customer and one trusted by the network, for example at the two points referenced (a) in the Figure. For ease of implementation, a single measurement point (b) trusted by both parties may be used. This might be implemented, for example within a secure module such as a cryptographic card on the client terminal. As an alternative, measurements may be made at different points with some possibility of discrepancies between measurements. On the network the practical measurement point is at the first access device(s) that, for each customer, inspects network layer headers (c)(IP in this case). ISPs should not measure any deeper into their network (d) because their access network and systems will introduce delays and losses.

For an individual customer (e.g. on dial-up access), a practical point at which to measure would also be alongside the network layer but in their end-system's stack (e). Ideally these measurement points would be lower in each stack to be closer to the interface between the two parties and less likely to be affected by contention in the stack. However, measuring at the link layer (f-f) would be inappropriate because only some chargeable parameters set at the network layer will ever be reflected in link layer frames; network level multicast, end-end latency requirements etc. may never be visible at the link layer. Also, link layer headers would need to be ignored when measuring packet sizes for bandwidth calculations to avoid apparent discrepancies where different link technologies are chained together.

In the reception direction (up the stack) this choice of measurement points implies that the lower layers must be dimensioned (buffer sizes, interrupt and thread scheduling priorities) to cope with the most stringent QoS requirements of higher layers. As frames are taken off the physical media, the machine must be able to pass data up the stack without any chance that usage-charged data gets discarded (e.g. due to buffer overflow caused by interrupt contention) before it gets to the network layer. It is at the network layer where the ISP's service is to be measured and where it is most convenient for QoS requirements to control correct differential treatment of the various flows as they are passed further up the stack (on end-systems) or forwarded (on routers).

The measurement objects described above may be implemented using, with appropriate modifications, publicly available network metering software such as Nevil Brownlee's NeTraMet system. This is a software meter which conforms to the IETF internet accounting architecture described in RFC 2063 and RFC 2064. The meter builds up, using "packet sniffing", packet and byte counts for traffic flows, which are defined by their end-point addresses. Although generally, Addresses can be ethernet addresses, protocol addresses (IP, DECnet, EtherTalk, IPX or CLNS) or 'transport' addresses (IP port numbers, etc), or any combination of these, in the present implementation IP addresses only are used. The traffic flows to be observed are specified by a set of rules, which are downloaded to NeTraMet by a 'manager' program. Traffic flow data is collected via SNMP (Simple Network Management Protocol) from a 'collector' program

Figures 5a to 5e are class diagrams illustrating an implementation of the measurement and accounting objects described above. The class diagrams are shown as a series of views.

The control view (5a) groups the classes related to control over the accounting class, including reporting control, metering-related control and general control functions. This view also relates to event dissemination. Control over the Accounting class is separated according to the type of control. This is why four interfaces are available. Two of those interfaces provide direct control over the behaviour of the Accounting object and the two others are related to a Java event model used to communicate both reporting information and measurement information. The ActControl interface provides control over the accounting class that relates to the accounting behaviour in general. It provides both methods to set a behaviour or properties and methods to find out about the current behaviour of the accounting object. For example, this interface is used to set the name of the accounting object or to query the Act object to find out a name previously given to the Act object. The ActReport interface provides control over issues related to account reporting. Control calls are directly related to the reporting behaviour of the accounting object. For example, a method named addReportListener() is used to register interest in reporting information. Once the registration is effective, subsequent calls to other control methods define behaviour such as the reporting frequency, request for immediate reporting, reporting security properties..etc. The two other listener interfaces (Report & Measurement) that the Accounting class implements are used to indicate that accounting objects are interested in accounting reports and measurements.

The accounting report view (Figure 5b) regroups the class related to the reporting behaviour and reporting process in the accounting objects. The accounting objects listens to accounting reports and generates such events as well. Accounting objects generate accounting reports and distribute them (using the traditional Java event model) to objects that have registered their interest in such events. In the present implementation flexinet (A CORBA like distributed programming infrastructure) is used to support communication between objects so that the reports may be from objects that are remote from the accounting object. The Accounting class implements the ReportListener interface so that it can receive those accounting reports as well. The accounting report events are of a ReportEvent class. An event in this class is a traditional Java event which includes a Report object. The main attribute in the Report class is records. Records is a simple vector of accounting records. These records are described in the AccountingStoreView. The ActReportCtrl interface defines the control calls related to the accounting reporting process of an accounting object. Calls are available for an object to register interest in accounting reports, de-register interest and to control the reporting process.

The accounting store view (Figure 5c) regroups the class related to the persistent storage of accounting information. An accounting object has a Database of accounting Records. The Record type holds accounting information which is not priced. Priced information is the subject of a different class. The Database class is a simple Vector of Record objects and it can be serialized to a file on a external storage medium. The database is also responsible for returning accounting records that have to be reported.

The accounting meter view (Figure 5d) regroups the class related to metering aspect of the accounting class. This relates both to the reception of the measurement information in the accounting objects and also to the control of the Meter as well as the definition of a simple Meter class. The Meter class uses a "Pulsar" object that generates pulses events as required. The frequency of pulses is set by the Meter object. On reception of pulses the Meter generates objects of type MeasurementEvent. Objects implementing the MeasurementListener interface and that have registered their interest in measurement results will then receive those events via a measurementHandler method. As previously noted, the Meter object and one or more of the objects receiving measurement events may be remote from each other. A measurement event is a conventional Java event and includes a measurement record of type MeasurementRecord. An accounting object gets measurement information from a Meter over which it has got control via the MeterControl interface. A typical example of control is the measurement reporting frequency, that is, an accounting object may control the frequency with which a meter object sends reprots to it. This control interface is also the one to use to register interest in measurement results.

The accounting miscellaneous view (Figure 5e) regroups all the other classes that do not fit in the previously described views. This includes, JavaBean-related classes, classes to run the code and graphical user interfaces (GUI). The AccountingBeanInfo class is a JavaBean related class which modifies the description of some attributes on the Accounting class when those properties have to be graphically displayed in the BeanBox or in any other component builder tool. The Go and MeterGo classes only implement a main method. Go is used to launch an accounting object and MeterGo a Meter object. The AccountingGUI class is responsible for the GUI related to the accounting objects. The Meter object has no GUI associated with it. The Accounting GUI is shown in Figure 6. The top part of the GUI includes data from the Accounting object and the bottom part relates to the control available over the accounting object. The control part is directly related to the control interfaces available for the Accounting objects. The accounting class is not aware of the GUI as the reference is from the GUI to the accounting class.

The accounting mechanisms described above can be used in combination with contracts between customers and retail and wholesale networks to establish liability to pay and who is expected to pay. The following section describes different clearing models for the making of payments. The systems described in this section may be used in conjunction with, or independently of the specific accounting mechanisms described above.

Payment Clearing

As well as "liability to pay" and "who is expected to pay" there is also the question of who should be paid. It is preferable for it to be customary for each edge ISP to be paid on a "half-circuit" basis for both their sent and received service. However other business models need to be considered. In particular, we will now consider a model similar to the public phone service, which has one or two implicit features that need to be separated out for full understanding.

Let us consider a business model where ISPs don't expect payment for all sent and received traffic to be made to all edge providers. Instead a customer might pay their own provider on behalf of both (all) ends as in telephony. A further accounting field would appear to be necessary - a "payee" field. For instance, this alternative business model might be that the decision as to which end(s) payment from edge customers entered the system was made on a per flow basis by customers. We shall call this model the "provider clearing" model for reasons that will become clear as we go. This is shown in Figure 9. Here, end customers 91,92 communicate via a number of intermediate networks 93. The financial flows between providers in this model depend on at which ends payment is entering the system on a per flow (or per packet) basis. For some flows, there may even be proportional sharing of costs between the ends. Therefore, for business model flexibility, rather than stating simply "local" or "remote" end, the "payee" field could be a "payee percentage" field instead - the percentage of the total cost to be paid by the customer at the end being accounted for. So usually it would be 100% or 0% in the typical cases of "paid completely to local provider" or "completely to remote". The balance would be the remote end's payment. Note, though, that the perceived purpose of this model is the transaction efficiency when the local payee gets 100%. However, there are certain disadvantages for the "provider clearing" model:

As already pointed out, the "payee percentage" field would have to drive inter-provider accounting, otherwise the revenue of an edge ISP and its upstream providers would depend on a factor completely outside their control - to which end its customers chose to make payment. The "payee percentage" field would therefore have to be trusted by upstream providers. To help prevent the field being tampered with, it would need to be signed by the remote ISP. How signed fields can be aggregated without losing the signature integrity is a matter for further research. The aggregation might have to be done by software signed by a third party trusted by all the parties involved (TTP) and then the record re-signed by the TTP. However the aggregation software would also have to run on a host trusted by the TTP. Further, using this model would mean that all edge ISPs would have to be able to identify any remote ISP from the remote address, something not possible with hierarchical routing. Nonetheless, we have already stated that the payment interface of the remote ISP can be passed in a higher level protocol between end stations. It would be only slightly more complex for them to include this in the accounting record. However, the ISP would still have to make appropriate checks that this was a valid ISP and that it matched the remote address. Once it has the address this becomes trivial, but more inefficient and rather negates the advantage of the local ISP doing the clearing via its upstream provider. Still further complication might be introduced for some future applications if the share of payment between the parties wasn't fixed but depended on characteristics of the flow or other parameters only understood at a higher level - higher than the provider would normally be interested in. This is also a problem for the "expected payer" field, but in that case the field is informational only, unlike the "payee percentage" field in the "provider clearing" model.

Worse still, the payment should ideally be split taking into account the current prices of all the edge providers who will eventually be paid.The only alternative (used in the international accounting rate system (IARS) for telephony) is for ISPs to agree compromise prices between themselves that average out price inconsistencies. This is what has been causing all the tensions in IARS as some countries liberalise earlier than others causing huge variation in prices around the world, between which no compromise can be found with which all involved are content. This is difficult even for a system where every end to end path only passes through two international carriers at maximum, each pair setting compromise prices with eachother. With nine ISPs on many end to end Internet paths and considerable peer interconnection, the horse trading would be a nightmare.

Finally, because of the much longer provider chains typically found on the Internet, potentially unacceptable delays will be introduced before the revenue arrives in the correct place. Any delay in clearing hugely increases the cost of the payment system, as extra trust mechanisms have to be invoked while the payment remains unconfirmed. These trust mechanisms have to be applied to the edge customers, not just the providers, therefore hugely increasing the total cost of the system.

Despite this limitations, the reason such a model is appealing is that it appears to reduce the number of payment transactions. For example, if the parties in an Internet 'phone conversation are both (all) being paid for by the caller, it appears less complex for the caller to pay everyone's payments to her own ISP, then let the ISP transfer the correct amount to its upstream provider as part of a bulk transaction. On the other hand, in a "third party clearing" model (shown in Figure 10), the caller has to split up the payment between both (all) ISPs of both (all)parties involved.

This is why the distinction between the names of the two models is in the clearing, not who is paid. Both models end up with edge ISPs paid on a half-circuit basis. The difference is merely in the route the payment takes from payer to payee. With provider clearing the payment follows the data path. Along the way, providers take their cut with two types of money sharing being mixed together: wholesale cut half-circuit sharing

In the "third party clearing" model, the clearing house rôle deals with the half-circuit sharing (including the straightforward price differences between the two ends) leaving inter-provider accounting to be purely about wholesaling. There is nothing to stop providers or customers assuming the clearing house rôle, but the accounting information model needs to be based on a third party clearing system to allow for the most general case. To clarify, whether the paying customer makes payment to a dedicated clearing house, direct to the ISP at the remote end or even direct to the remote customer so that they can pay their own ISP , in all cases, the rôle of clearing must be separate even if there is no separate enterprise to achieve the function. Note that the last case is special - the clearing rôle is null, but it still appears in the information model. In other words, the charges for all ends should never be lumped together while accounting. If the half-circuit sharing is achieved through the provider chain, this must be kept separate from the accounting for wholesale. If it is not, the types of model that can be built on the infrastructure are restricted.

Having separated out the rôle of clearing, this now shows explicitly that a telephone company also bundled another rôle in its business- that of "session retailer". That is, the edge telco is offering telephony sessions at fixed prices, but the range of prices is less than the number of possible ways the price could vary if it were simply composed of all the end to end prices charged by providers necessary to assemble each session. Again, this rôle may be assumed by the edge customer in the Internet world, but it is possible that businesses will spring up offering prices for transmissions by selling on IP service while absorbing variations across providers in the prices they are charged wholesale. Obviously, this rôle may also continue to be taken by telcos and ISPs too.

It is redundant to state in accounting messages which end will actually be paid. Who should eventually receive the payment is implicit because the rule is now that accounts for other providers shouldn't be lumped with accounts for the local provider. The corollary is that any accounting implicitly relates to payments that will eventually end up with the local provider. Saying who will be paid is meaningless during accounting. It is only relevant at the time of payment. Then it essential to say who the payment is eventually intended for if it is given to a clearing organisation.

Other aspects of the invention are described, by way of example, in the annexed papers.

The Direction of Value Flow in Connectionless Networks

Abstract

{Re-write at end}
This paper proposes solutions to an apparently fundamental question in networking: "In which direction does value flow in a connectionless communications network?" Value flow is considered both between the ends of a communication and between the networks along the path of a communication. Multicast and aggregation modes are considered as well as unicast. The goal is to derive an optimal default business model for Internet service providers if charging for transmission. The search is for the most likely common case for apportioning the value of transmission between senders and receivers, in order to reduce the cost of dealing with exceptions to the norm. But this has to be reconciled with conflicting issues of blame, liability and control. A pre-requisite is to unravel the confusions that are common where higher level issues become embroiled with networking issues. The result is the creation of two possibly novel business models.

Keywords

Charging, pricing, end-to-end, Internet, business modelling.

1. Introduction

{Re-write at end}
Traditionally, data communications has been sold so cheaply that charging for it on usage basis has not seemed feasible or sensible. Where flat-rate subscription prevails, the question of the apportionment of the value of a particular communication between its ends rarely surfaces. With the possibility of variable quality of service (QoS) approaching, the need for some form of usage-charging for high QoS service has arisen. This has led to new thinking on cheaper usage-charging systems for packet networks [Clark96, xxxb99]. This has brought the issue of the apportionment of the value of a communication into the limelight.

Clearly, the cost of apportioning charges is significant, therefore it is important that the default apportionment matches the most common case. This paper establishes that default case then goes on to suggest a new role in the communications industry for apportioning charges between end-customers separately from the question of the apportionment of payment to network providers. This is contrasted with the current way apportionment is universally achieved in the communications industry and with some of the proposals in the literature.

{Talk about when it is worth bothering.}

The asymmetric nature of the relationship between sender and receiver is also discussed, with respect to blame, liability and control over the flow of information.

2. Related Work

Some authors state that they believe the busines model of the current fixed access rate Internet model is "sender takes all" [ITU96, Zull97]. This phrase is used to imply the sender's ISP receives all the revenue. Clearly, this is the exact opposite of the case. Traffic exiting a network uses the same sort revenue. Clearly, this is the exact opposite of the case. Traffic exiting a network uses the same sort of bandwidth as traffic entering. Given ISPs charge for bandwidth, both the sender's and receiver's ISPs receive revenue from any packet transmission. This is a similar position to the half-circuit charging common for data links, but applied end-to-end. MacKie-Mason *et al* asserts that blame is impossible to determine at the network level [MacKieVar92], an argument that can descend into sophistry, as both concepts are difficult to define. However, later, using precise definitions of the terms, we argue that the sender is always to blame for a transmission in a connectionless network.

Clark analyses the apportionment of charges between senders and receivers [Clark96], and proposes an engineering solution, which it is admitted would introduce considerable complexity to the Internet if implemented. Shenker *et al* describes edge pricing [Shenker96], a business model that is prevalent in communication networks and which forms much of the background to this work.

3. Assumptions

We use the 'black box' network routing assumption described in XXX [xxxb99]. Briefly, this means we trust a network or inter-network of federated domains to find the cheapest route without end-system intervention. If there is a cheaper route, the discrepancy is so minor that the end-customer isn't concerned. This also implies that internal network design inefficiency should be absorbed by providers in their overall pricing.

For instance in current multicast routing, tree stability always takes priority over tree efficiency . Providers are free not to make this decision if they can design better multicast routing algorithms. This approach consciously doesn't exactly share the cost of the multicast fairly based on the topology of each receiver's leg of the tree - a problem Herzog *et al* [Herzog95] analysed (although it didn't consider inter-domain models). Our approach gives a scenario where receivers are divided into sets based on domains that can each be priced independently. Similarly, senders can be priced separately. Thus, because distance is relatively unimportant, we assume it is sufficient to be able to charge for multicast on a per-domain basis. Just as XXX suggests for unicast, if distance related charging were required, pricing could be differentiated for various combinations of sender and multicast address, but we believe this is unlikely.

We also use the minimalist business model of a network provider introduced in XXX. That is, one that is only concerned with offering services at the local interfaces to their network layer business: they but in link layer services and higher layer services like domain name service.

4. The value of place

The value of communication concerns the incremental value of having information in a certain place or places, instead of or as well as the original places. Usually, the more the information is worth, the more value is placed on having it in the right places. There is no value to the customer at all while information is in transit. It is delivery that is important. Strictly one also has to take account of the mitigating cost of storage in both places (or only in the second place if the sender deletes after sending). In summary the added value of transmission is the marginal change in value caused by associating a new location with the intrinsic value of the information to the customer.

However, because the data communications market is fairly competitive, charges for communicating information tend instead to follow the 'cost plus margin' rule. This is particularly so because it is very difficult for providers to predict what value their customers put on moving any one piece of information anyway.

Any payment to an edge-network provider has the two aspects - 'who pays' and 'who is paid'. 'Who is paid' can only be each local provider collecting its local price. With competitive 'cost plus' pricing there is no scope for any provider to break out of that. But, because communications naturally involves at least two parties, in order to cover the total costs of all the providers involved, 'who pays' can be on a different apportionment. The edge customers *do* know the value to them of having the information at a certain place. Thus, although apportionment is difficult for network providers, it is very relevant to the edge-customers. However, clearly, the network providers can stimulate more use of their networks by making arrangements for customers to efficiently apportion costs between themselves.

5. End-to-end pricing

If a price is higher than the perceived value for any customer, she is free to get the remote party (or anyone else) to make up the difference through some higher level arrangement. On the other hand, if the value to her is higher than her local price, she is also free to offer to cover some of the costs of the remote end(s). However, our minimalist provider doesn't have to be concerned with matchmaking multiple customers to get round local discrepancies between price and customer value. This is an issue that can be dealt with end-to-end, not locally. We are not saying ISPs shouldn't offer end-to-end pricing - it is clearly in their interest to matchmake between customers with surplus value and those with deficit. All we are saying is that, if they do, end-to-end pricing should be considered as a separate role (Fig {3.1?}). Such a role could be a separate business - it could gain on some combinations and lose on others, possibly making a profit overall. In this case it would be a retail service that used the networking services as wholesalers. It is also possible that edge customers could effectively take on this role themselves. Fig {3.1?} shows two end customers using a data path through multiple connected ISPs. The relative value of the service flows and prices for one direction of one class of service is represented by the thickness of the arrows. Note that the size of the proportions of prices represents a choice by the end-system that is willing to pay more than its local price. Pricing between providers is omitted for clarity (but see later).

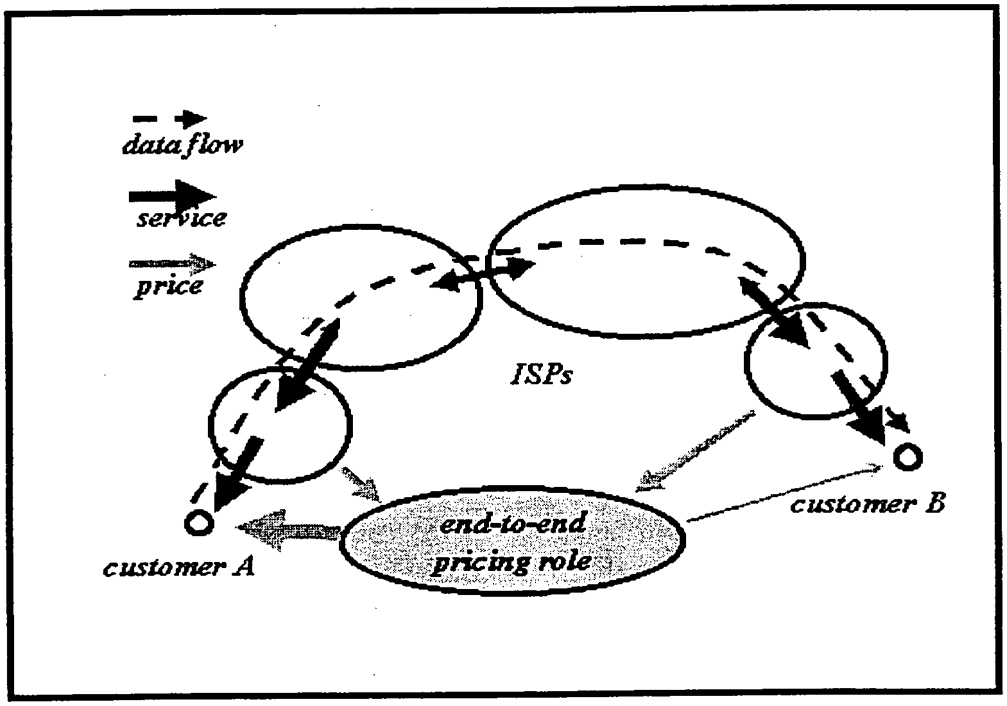


Fig {3.1?} - 'End-to-end pricing' role

Telephony firms have traditionally offered end-to-end pricing because they are selling an application. The role of network provider has always been muddled with selling the end-to-end application. This is already putting considerable strains on the International Accounting Rate System (IARS) [ITU_RIARS] with potentially $s(n-1)^2$ prices having to be negotiated (where n is the number of edge providers and s is the number of global schemes for sharing the proportions of the price between the ends, e.g. local rate only, free to sender). In practice, end providers are grouped together to reduce the number of prices presented to customers. The PSTN uses addressing conventions (e.g. +800 for free to sender), but this limits commercial flexibility to the few schemes that are widely recognised. Clark proposed a solution to allow flexibility [Clark96]. However, catering for various combinations of sender and receiver payments through the core of the network needs packet format changes and router involvement. Further, wholesale prices between providers would have to be negotiated for every possible scheme for sharing charges between the two ends as well as for every possible grouping of end points beyond that boundary. Worse still, inter-provider accounting would then require traffic flows to be isolated then further sub-classified by how much each end was paying on a per-packet basis.

The 'n^2 problem' would still exist for our end-to-end pricing solution but this is fairly easy to conta by grouping. {An example scenario is given at the end of the paper.} Importantly though, end-to-en pricing gets rid of all the inter-provider problems described above. There becomes no need at all to identify end-to-end flows at inter-provider boundaries. Thus inter-provider charging could be based on bulk measures like average queue lengths, routing advertisements etc. Also, most importantly, end-to-end pricing can be introduced without changing the Internet at all, and it allows future flexibility. To summarise so far, we should ensure any discrepancy in the willingness to pay across the end customers is normalised end-to-end first, so that edge ISPs always receive payment at their local price.

6. Common case value apportionment

Although we have delegated the problem of combining sender and receiver payments to a higher level, we propose that it is important to cater for the common case at the network charging level so that the higher level functions are unnecessary in most cases. The large majority of communication occurs between consenting parties. Therefore we propose that all edge providers should charge their local customers for both sending and receiving. Allowing different prices for each direction allows for asymmetric costs (e.g. access technology like ADSL or satellite) and for asymmetric demand (e.g. some ISPs might host more big senders, while others might host the mass of receivers). More specificallay, if operating usage-based charging, we propose a provider should aim to offer each class of packet transmission service in each direction at a separate price. There is obviously nothing to stop any two prices being the same. Classes of service are defined as a unique combination of their service mode (unicast, multicast) and their quality (latency, instantaneous bandwidth, reliability, jitter). Quality specifications within one class may leave one parameter to be specified by the customer while others remain fixed.

Now that we have eliminated all but local pricing, we show that we can extend this model recursively to apply at the boundary between any pair of providers. This becomes a generalised model of edge pricing whether the 'edge' is really at the network edge or just the edge of a backbone. Considering the cases of all edges - between providers and those at the edge customer - we now consider whether any one choice of apportionment between sender and receiver payments is more stable.

Fig {3.2?} shows a generic scenario with multiple networks, N, all connected to the network of interest, N_b. For each class of service, each connected network has a status relative to N_b based on whether it provides more or less connectivity to other hosts at that class of service. Although the diagram gives the impression that N_b is a backbone network, any one of the neighbouring networks could be a simple link to an edge customer's single host. The model is designed to be general enough for N_b to be an edge customer, an edge network, a backbone network or some hybrid. Those networks with the same suffix are of similar status relative to N_b. For instance, those labelled N_c may be edge customers, N_d may be equally large backbones and N_e a peer network.

A packet is shown being multicast from N_a into N_b and onward into the other networks. Because multicast is a general case of unicast this allows us to model both topologies. We will also be able to treat the topology as aggregation(i) by reversing the direction of transmission. The term packet is used, but the arrows could represent flows of similar class packets for a certain time. The packet or flow being modelled could be data or signalling. It is not necessary to model multi-source multicast separately because packets from different sources always remain separate. Fig {3.2?} highlights the pricing between networks N_a and N_b. W_{bas} and W_{bar} denote the per direction weightings applied to the charge that N_b applies to N_a. W_{abs} and W_{abr} likewise weight the charge N_a applies to N_b. Each weighted price is for transmission between the edge in question and the remote edge of the Internet, not just the remote edge of that provider. There would be four price weightings like this for every class of service at every inter-network interface, but the weights would take different values unless the neighbours were of the same status. We call this model 'split edge-pricing'. The relationship between any two parties across the edge of their networks is split into prices for each class of service and each of these is further split into two prices for each direction, each of which are again split into 'half' prices that each party offers the other.

{insert diagram of multi-layer pricing with one class of service for one pari becoming two for another pair and discuss?}

Thus the payment for traffic in any one direction across each interface depends on the difference between the two weighted prices offered by the networks either side. In other words, no assumptions are made about who is provider and who is customer; this purely depends on the sign of the difference between the charges at any one time. Clearly, edge customers (N_c, say) have no provider status in the networking market. So, for all j, $W_{cjs} = 0$ and $W_{cjr} = 0$. We can then analyse scenarios like 'only senders pay' or 'only receivers pay' by setting all receiving weights to zero or all sending weights to zero. For instance, stability of a policy can be determined by assessing whether one network would gain from a maverick policy.

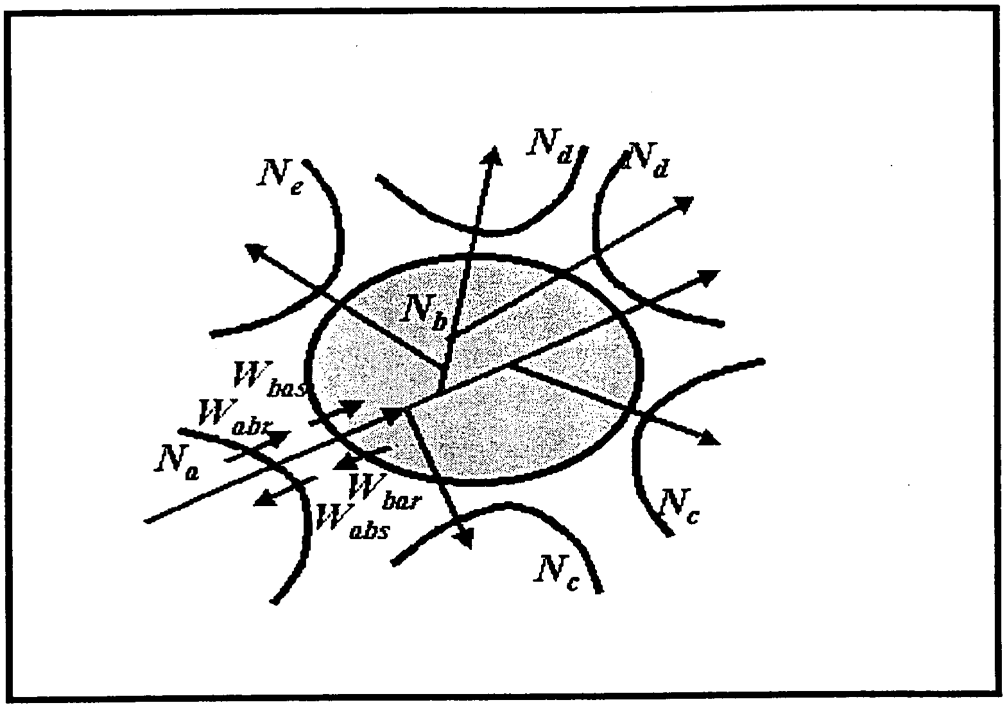


Fig {3.2?} - Charge for sending, receiving or both?

'Only senders pay' or 'only receivers pay' tends to encourage migration of customers who are primarily receivers and those who are primarily senders to different providers. This situation is tenable because the provider with all the non-paying customers gets all its revenue from its interconnect business. Either scenario remains stable, because if one network goes maverick (e.g. only charges receivers when everyone else is only charging senders), both predominant senders and receivers have a choice of cheaper provider. Therefore the income to the whole system reduces ensuring the maverick provider would go bust first - sufficient disincentive to be maverick! However, either of these policies clearly make network utilisation inefficient with more traffic crossing the inter-provider boundary.

Both policies are unstable where multicast is concerned. With 'only multicast senders pay' there is no way to ensure the networking costs of all receivers are covered without ruining the assumption of IP multicast that the sender should be unaware of the number of receivers joining. This assumption is fundamental to the scalability of receiver initiated multicast. Hence the sender's ISP will also be unaware of the majority of receivers in a multi-domain scenario and no-one downstream has a financial incentive to report this. 'Only multicast receivers pay' simply leads to all multicast receivers migrating to the first maverick provider who operates 'only multicast senders pay', even if the latter is offered with some random guess at the sender price level for a common number of receivers on a tree. This makes 'only multicast receivers pay' a recipe for bankruptcy.

In contrast, 'both senders and receivers pay' is stable in both unicast and multicast cases. It also doesn't lead to inefficient network utilisation unlike the above cases. It is also possible to cater for diff nt balances of predominant senders and receivers by weighting the sending price differently to the ıeceiving price. For instance if there are a few big predominant senders but many small predominant receivers, the economy of scale in managing a large customer can be reflected in a lower sender weighting. Similarly, the inefficiencies of multicasts to small receiver communities compared to multiple unicasts can be discouraged by slightly weighting multicast sender pricing. The aggregation case is similar, with 'both senders and receivers pay' stable while the two other policies go unstable for the same reasons as for multicast, but swapped round.

7. Blame, liability and control

We have shown that all ends paying is both the common case and a stable one so should be the default. We can share the cost differently at higher level if end user value is shared differently from this default (and it is worth bothering given the cost of another financial transfer). However, we must remember that a sender can decide not to send but a receiver can not avoid being sent to (in the current Internet). Clearly, if someone operates a Web service, they don't normally decide whether to send replies on a request by request basis. But this doesn't mean they have been forced to send. They have chosen to put the service on a well-known port with public access. They can only stop certain people requesting them to send by securing the Web server or interposing a firewall. If they want to block someone, sending an error reply is courteous, but it is not mandatory to be polite.

Ultimate sender blame presents a problem. In cases where the sender derives surplus value from a communiation and the receiver derives less value than their provider charges, receivers are vulnerable to being exploited. Such cases are much rarer than it first appears, mainly because of confusions that can be cleared by considering the following factors:

- The value of the information isn't relevant when considering the networking service - only the value of *moving* the information - getting it to a useful place
- Often the value of moving information is transitory - getting it to a useful place to discover that moving it wasn't useful
- Often the value of moving lots of information is to get a small part of it to a useful place, but it isn't possible to know which part before moving it
- The cost of transmitting information is often far less than the cost of targeting which information should be transmitted
- Information in one direction often controls the flow of information in the other

Nonetheless, genuine cases remain where the receiver is being persistently forced to pay for transmission that is valuable to the sender but not to the receiver. The only solution to this seemingly intractable dilemma is for it to be *customary* for all ends to pay, but the ultimate liability should remain with the sender. Any receiver could then dispute the customary apportionment (end-to-end) with no risk of denial (unless the sender had proof of a receiver request). A similar but opposite situation used to prevail with the UK postal service. It was customary for the sender to pay for the stamp, but if it was missing or insufficient the receiver was liable for the payment, because the Royal Mail had an obligation to deliver every letter.

8. End-to-end clearing

We have discussed how prices can be apportioned between the ends of a communication. We now discuss how payment will follow the same path. We can assume electronic commerce will make it possible for anyone to pay anyone else's ISP on the Internet, even if a clearinghouse is needed We shall call this the 'end-to-end clearing' model (Fig {4.2?}). These arrangements will typically be made through higher level protocols. The act of making a financial transfer has similar order costs to the cost of transmission of a couple of e-mails. In addition there will be local processing costs for authentication. Therefore, arranging a different apportionment of charges between ends is more likely for long-lived sessions, such as Internet telephony or conferences on the mbone, than short connections, such as are typically on the Web. However, a collection of related short connections may be combined into one longer lived session for these purposes.

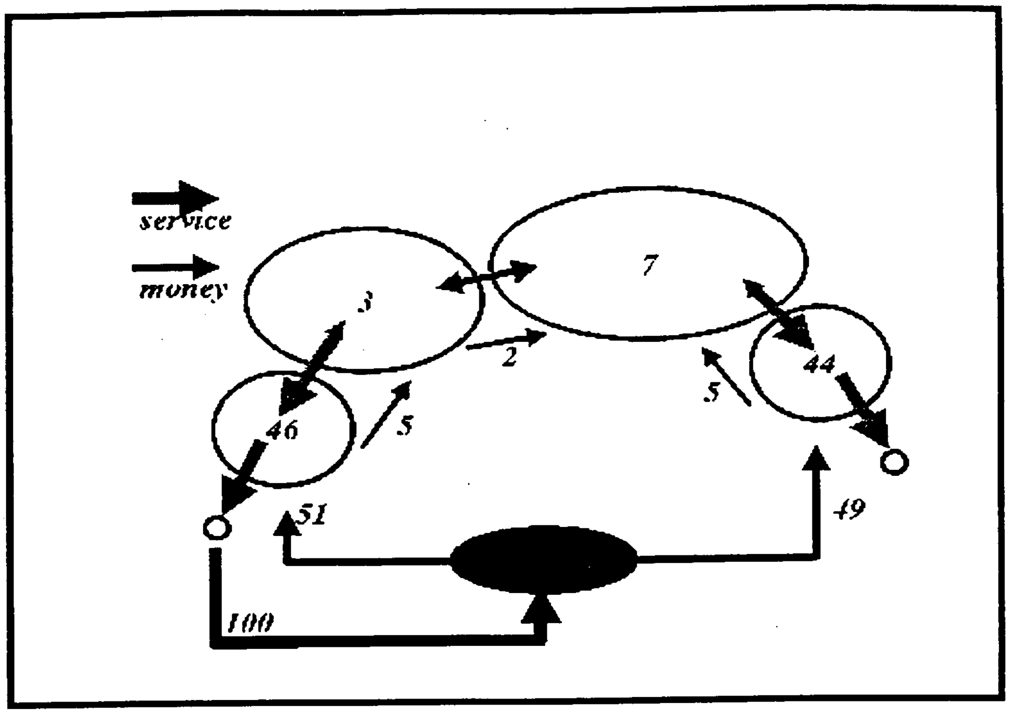


Fig {4.2?} - 'End-to-end clearing' model

In the 'end-to-end clearing' model, the clearinghouse role deals with the half-circuit sharing (including the straightforward price differences between the two ends) leaving inter-provider accounting to be purely about wholesaling. The figure shows one end paying and follows example percentages of this money as they are distributed among the providers. The value flows of the networking service relate directly to the money flows in the opposite directions.

There is nothing to stop providers or customers assuming the clearinghouse role, but the accounting information model needs to be based on a third party clearing system to allow for the most general case. To clarify, the paying customer may make payment:

- to a dedicated clearing house
- direct to the ISP at the remote end {(the remote customer need only notify her ISP's payment interface to the payer)}
- or even direct to the remote customer so that they can pay their own ISP

In all cases, the *role* of clearing must be separate even if there is no separate enterprise to achieve the function. Note that the last case is special - the clearing role is null, but it still appears in the information model. In other words, the charges for all ends should never be lumped together while accounting. If, instead, half-circuit sharing were achieved through the provider chain, end-to-end clearing information would have to be identified separately from that needed for wholesale accounting. If it were not, the types of model that could be built on the infrastructure would be restricted.

9. Iterative model

We have presented what we believe to be an optimum model, but other business models need to be considered. In particular, we will now consider a model similar to the public 'phone service, which has one or two implicit features that need to be separated out for full understanding. We will consider payment in the model first, rather than pricing, as it will then be easier to understand the pricing issues.

In this model, ISPs don't expect payment for all sent and received traffic to be made to *all* edge providers (Fig {4.1?}). Instead a customer might pay their own provider on behalf of both (all) ends as in the normal case for telephony and as proposed by Clark for the Internet [Clark96]. This alte tive business model might allow the decision as to which end(s) payment from edge customers entered the system to be made on a per flow basis, by customers. We shall call this model the 'iterative' model for reasons that will become clear as we go. The financial flows between providers in this model depend on at which ends payment is entering the system on a per flow (or per packet) basis. For some flows, there may even be proportional sharing of costs between the ends. For business model flexibility an accounting system would need a 'payee percentage' field - the percentage of the total cost to be paid by the customer at the end being accounted for. Usually it would be 100% or 0% in the typical cases of 'paid completely to local provider' or 'completely to remote'. The balance would be the remote end's payment. Note, though, that the perceived purpose of this model is the transaction efficiency when the local payee gets 100%.

If Fig {4.1?} is compared with the end-to-end clearing model in Fig {4.2?}, both models end up with edge ISPs paid the same amounts on a half-circuit basis. The difference is merely in the route the payment takes from payer to payee. With 'iterative' clearing the payment follows the data path. Along the way, providers take their cut with two types of money sharing being mixed together:

- wholesale cut
- half-circuit sharing

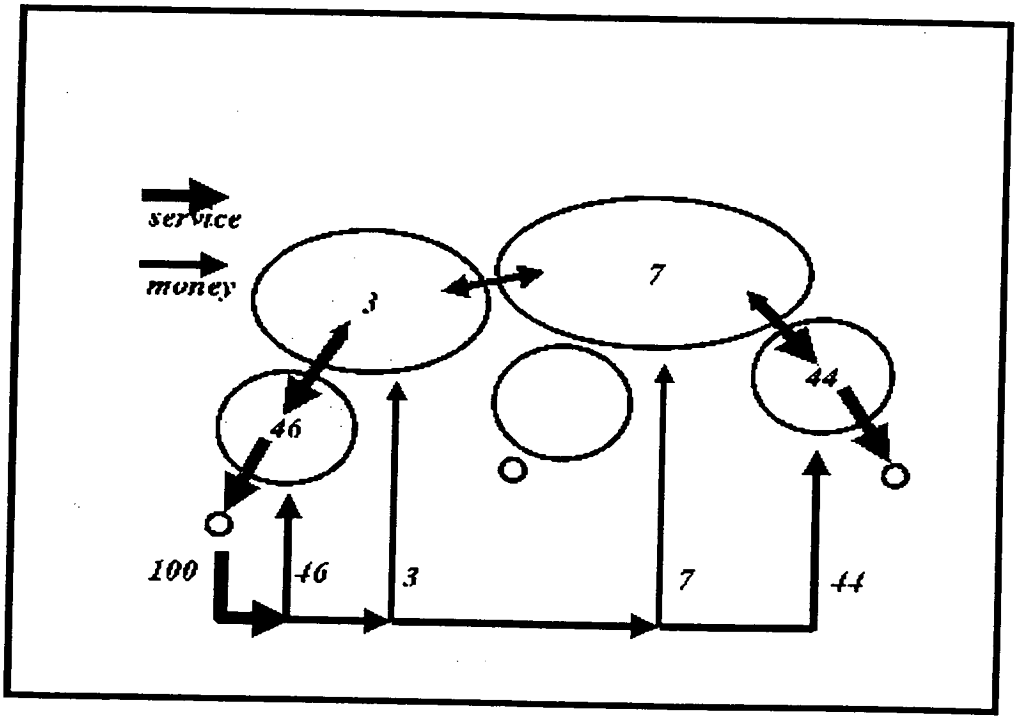


Fig {4.1?} - 'Iterative clearing' model

However, the amount deducted from the flow at each boundary doesn't match the level of service crossing that boundary. This can lead to complexity in the network, as there is pressure to design the network itself to reveal the apportionment of costs. This was why Clark was concerned about how much complexity would be added to the Internet to cater for arbitrary combinations of sender and receiver payments. This is also why international and interconnect on the PSTN have limited flexibility to arbitrarily apportion charges between the ends. Even free to sender calls are blocked between a lot of countries because they don't yet have prices set. Specifically, there are five points stacked up against the 'iterative clearing' model:

- As already pointed out, a 'payee percentage' field would have to drive inter-provider accounting, whether it was in accounting messages or packets. Otherwise the revenue of an edge ISP and its upstream providers would depend on a factor completely outside their control - to which end its customers chose to make payment. The 'payee percentage' field would therefore have to be trusted by upstream providers. To help prevent the field being tampered with, it would need to be signed by the remote ISP. How signed fields can be aggregated without losing the signature integrity would be a matter for further research.

Still further complication might be introduced for some future applications if the share of payment between the parties wasn't fixed but depended on characteristics of the flow or other parameters only understood at a higher level - higher than the provider would normally be interested in.

- Worse still, the payment should ideally be split taking into account the current prices of all the edge providers who will eventually be paid. The only alternative (used in the international accounting rate system (IARS) for telephony) is for ISPs to agree compromise prices between themselves that average out price inconsistencies. This is what has been causing all the tensions in IARS as some countries liberalise earlier than others causing huge variation in prices around the world. between which no compromise can be found with which all involved are content. This is difficult even for a system where every end to end path only passes through two international carriers at maximum. each pair setting compromise prices with each other. With eight ISPs on many end to end Internet paths. five typical [McCreary98] and considerable peer interconnection, it is likely that it will take longer to negotiate prices than the time available, thus leading to distortions to providers' supply and demand signals.
- Finally, because of the much longer provider chains typically found on the Internet, unacceptable delays will be introduced before the revenue arrives in the correct place. Any delay in clearing hugely increases the cost of the payment system, as extra trust mechanisms have to be invoked while the payment remains unconfirmed. These trust mechanisms have to be applied to the edge customers, not just the providers. therefore hugely increasing the total cost of the system.
- If multicast is to be catered for by iterative clearing (e.g. conferences). each provider needs to know how many ends they are serving locally, both to inform the person paying and check settlement. In the end-to-end model, only the ends need to know how many ends there are to pay for - no-one needs to calculate how many ends are attached to each provider. Thus, for instance, if there is a charge to join a conference, this can cover the cost of paying each participants' communications charges. The more who pay the host to join, the more there is to cover charges.

The only advantage of the 'iterative' model is that it appears to reduce (by one) the number of transactions to achieve the desired apportionment. Also all the inter-provider transactions can be fairly lightweight because they can be batched up. For example, consider the case where both the parties in an Internet 'phone conversation are being paid for by the caller. It appears less complex for the caller to pay everyone's payments to her own ISP, then let the ISP transfer the correct amount to its upstream provider as part of a bulk transaction. However, on the other side of the bargain is a considerably more complicated network, compromise pricing, increased credit time lags and less flexibility in inventing new ways to apportion charges, particularly for multicast.

10. Example scenarios

10.1 Finding an end-to-end price

Let us assume some way has been invented for an ISP's edge custome, C_a, to announce her intention to cover some part of the transmission costs of parties communicating with her, C_b, C_c etc. Some suggestions are given in [Clark96]. Kausar suggests modifications to SDP [RFC2327] to achieve this for longer sessions [Kausar99]. A price needs to be set and settlement made between the parties. If this is achieved, end-to-end, between the parties involved there are no further engineering implications - the two parties clearly trust each other enough to enter into a financial arrangement and are willing to accept the cost of the transaction. However, there will be many occaisions where the two parties have no trust relationship. In these cases the problem reduces to, C_b, C_c etc finding suitable intermediaries. First they must know C_a's ISP. C_a may have already given this information in some higher layer protocol. Alternatively a directory of ISPs could be operated by the Internet address allocation registry (IANA) or any private concern, in which one could look up the network address of C_a and be given the payment interface of the associated ISP. This directory might be operated by one organisation monolithically or or it could be hierarchical like DNS. They may then choose to check whether their own ISP has a direct relationship with C_a's ISP. Alternatively, they could go straight to another directory we postulate would be necessary. This directory would accept lists of ISPs and return a list of organisations that would act as an intermediary between them all. The mechanism would be identical to a Web search engine - in invert index of ISP-intermediary pai· hat would accept queries of logically AND'ed ISPs.

The resulting intermediary could then be contacted to find the price being offered for the particular combination of ISPs. The same organisation would naturally take the payments and clear them between providers.

10.2 Separating accounting and settlement

We must remember that we concluded earlier that only the sender should be ultimately liable for usage charges. However, we suggested that the customary position should be to expect every customer to pay for both reception and sending. At this point we have to make a clear separation between accounting and liability for payment. We propose that each customer provider pair should first reconcile their *accounts* for both sent and received traffic. Even if traditional billing with no reconciliation is being used, we are implying the local customer should be sent an *account* for all her traffic in both directions. The customer immediately acknowledges, declaring which of the received traffic she is willing to be liable for. This will leave a list of usage records to be paid for by the sender, including the sender's address on each, of course. For the provider's information, she may identify which records she believes the remote party expects to pay and which she is simply disputing. She is already liable for all the sent traffic, of course.

10.2 Inter-domain multicast with heterogeneous QoS

Illustrating the power of the principles set down so far, we can take an example level of service like multicast with heterogeneous QoS per receiver and show that charging for it with correct apportionment will 'just happen', even inter-domain. However, it will only be efficient by using the 'split edge-pricing' model. For illustration, let us have two provider networks with an edge customer of N_a sending into a multicast address where the tree crosses into N_b reaching receivers who are customers of N_b. N_b will have given a price for multicast reception and N_a one for sending to an address in the multicast range. The tree may spread to other receivers on other networks too. The receivers in each domain will note when they join the multicast and each start being charged for the traffic they receive at their local price. The sender will be charged at her provider's price. At the domain boundary between N_a and N_b, N_b will be charged N_a's price for sending to a multicast while N_b will charge its price to N_a for receiving from a multicast. This usage may either be measured exactly at the inter-provider border, calculated statistically by combining customer usage data with multicast routing tables or simply covered within bulk measurements at the border. The receivers of a particular multicast group may happen to be located so that the multicast tree fans out immediately at its entrance to the network. With heterogeneous QoS per receiver (e.g. RSVP), any message to set up the QoS must emanate from the receiver and can therefore be charged for locally. Again this can be treated identically at the inter-provider boundary.

We believe edge pricing allows enough flexibility to charge differentially for broad ranges of route lengths because it allows different charges for different administrative domains. Even if a single domain spanned the globe, if desired it could be divided into internal pricing domains to achieve the same effect.

{Go on to discuss end-to-end apportionment}

10.3 Phone to Internet gateway

TBA

11. Limitations and Further Work

TBA

12. Conclusions

We have shown that the common case for apportioning value between the ends of a connectionless communication network is catered for if all users pay for both sending and receiving. We have also shown that this is the most stable and efficient case, particularly for multicast and aggregation. It should therefore be the default apportionment for payment purposes.

We have suggested that a new business model would be useful and more efficient to cater for the cases where there is a large discrepancy from this default in terms of value apportionment {...} - large enough for it to be worth making a balancing transaction given the costs of doing this. This new model requires a new role in communications markets - an end-to-end pricing role. In discussing clearing of payments across an end-to-end path, there is also a need for a third party role for end-to-end clearing. These two roles only make sense as new types of business if they are enacted by the same business. Otherwise customers will be paying money to a different organisation than the one quoting prices, which has obvious security flaws.

This new role could be conducted by existing ISPs or customers themselves, but there appears to be considerable added value, making this a viable business in its own right. It appears that this role is a threat to existing ISPs business. This role turns edge ISPs into wholesalers for a potentially large class of Internet applications. The end-to-end pricing and clearing role would become the retail face of the Internet in many cases.

Further, we suggest a subtle twist to the recommendation that customers should pay for both sending and receiving. We suggest this should be customary, but that ultimate liability for sending should lie with the sender. Disputes could then quickly be resolved through the end-to-end clearing role.

{search and destroy confusions between the words layer, level and class}

13. Acknowledgements

{Richard Gandon, BT International Carrier Services}

14. References

[McCreary98] Sean McCreary and kc claffy, "How far does the average packet travel on the Internet?", CAIDA, 25 May 1998, <URL:http://www.caida.org/Learn/ASPL/>

[Clark96] David D Clark (MIT), "Combining Sender and Receiver Payments in the Internet", in Interconnection and the Internet. Edited by G. Rosston and D. Waterman, Oct 1996, <URL:http://diffserv.lcs.mit.edu/>

[Finlayson98] Ross Finlayson, Discussion at Third Reliable multicast research group meeting, Orlando, FL, and following on the mailing list, Feb 1998 <URL:http://www.east.isi.edu/rm/>

[ITU96] ITU, "The Direction of Traffic", ITU/Teleography Inc, Geneva, 1996 <URL:http://www.itu.ch/ti/publications/traffic/direct.htm> in brief chapter 3, Box 3.1 is extracted on-line: "Accounting rates and how they work", <URL:http://www.itu.ch/intset/whatare/howwork.html>

[ITU_RIARS] ITU "Reforming the International Accounting Rate System" <URL:http://www.itu.ch/intset/>

[Kausar99] Nadia Kausar, Bob Briscoe and Jon Crowcroft (UCL), "A charging model for Sessions on the Internet", submitted to {??? IEEE}, Egypt, {??} 1999, <URL:{?}>

[MacKieVar92] Jeffrey K MacKie-Mason and Hal Varian, (UMich), "Some Economics of the Internet', Tenth Michigan Public Utility Conference at Western Michigan University, March 25-27, 1993: <URL:http://www.sims.berkeley.edu/~hal/people/hal/papers.html>

[RFC2327] Mark Handley, Van Jacobsen, "SDP: Session Description Protocol", IETF RFC 2327, Mar 1998, <URL:rfc2327.txt>

[Shenker96] Scott Shenker (Xerox PARC), David Clark (MIT), Deborah Estrin (USC/ISI) and Shai Herzog (USC/ISI), 'Pricing in Computer Networks: Reshaping the research agenda', SIGCOMM

Computer Communication Review Volume 26, Number 2, Apr 1996, <URL: http://www.statslab.cam.ac.uk/~frank/PRICE/scott.ps>

[Speakman98] Tony Speakman, Dino Farinacci, Steven Lin, Alex Tweedly, (cisco) "PGM Reliable Transport Protocol Specification", Work in progress: IETF Internet Draft, Jan 1998 (expires Jul 1998), <URL:draft-speakman-pgm-spec-01.txt>

[xxxb99] XXX, "Lightweight, End to End, Usage-based Charging for Packet Networks", Submitted to ACM SIGCOMM'99 , Jan 1999, <URL:yyy> (attached)

[Zhang93] Lixia Zhang (Xerox PARC), Stephen Deering (Xerox PARC), Deborah Estrin(USC/ISI), Scott Shenker (Xerox PARC) and Daniel Zappala (USC/CSD), "RSVP: A New Resource ReSerVation Protocol", IEEE Network. Sep 1993. <URL:http://www.isi.edu/div7/rsvp/pub.html>

[Zull97] Chris Zull (Cutler & Co), "Interconnection Issues in the Multimedia Environment", Interconnection Asia '97, IIR Conferences, Singapore, Apr 1997 <URL:http://www.cutlerco.com.au/core/content/speeches/Interconnection%20Issues/Interconnection

---

Notes

(i) Examples of packets that are forwarded until aggregation (reverse multicast) are:

- RSVP[Zhang93] receiver initiated reservation (RESV) messages
- pragmatic general multicast (PGM) [Speakman98] negative acknowledge (NACK) messages or the "lay breadcrumb" messages[Finlayson98] suggested in their place

Lightweight, End-to-end, Usage-based Charging for Packet Networks

Abstract

This paper suggests that a multi-service packet network might be achieved by adding classification and scheduling to routers, but without policing. Instead, a lightweight per-packet charging system is proposed that could fulfil the policing function and is completely separated from the data path. A high proportion of charging operations run on customer systems to achieve this, the proportion being configurable per-customer. Functions dispersible to customers include not only metering, accounting and billing but also per-packet or per-flow policing and admission control. Lower cost is achieved through simplicity without sacrificing commercial flexibility or security. Inter-provider charging, multicast charging and bundling network charges with those for higher class services are all catered for within the same, simple design. The paper is primarily architectural, but also reports on early implementation experience in an Internet context.

Keywords

Charging, pricing, congestion control, quality of service, admission control, operational support, active networks, end-to-end.

1. Introduction

The ideas in this paper can be adopted at a number of levels. At one level, the paper simply offers a very cheap way of charging for multi-service packet networks by moving nearly all operations to customer machines. At the most ambitious level, it completely removes the need for policing from a multi-service network. This means a full multi-service packet network might be achieved with the complexity of policing and charging completely separated out except at the end-systems. This leaves the network infrastructure clear to simply classify, route, schedule and forward. If classification is based on the simple differentiated services model (diff-serv) [RFC2475], this implies no need for flow-related state in the network at all. However, this doesn't imply the ideas will only work if adopted by all network providers. The solution is intrinsically designed to inter-work with non-usage-charging approaches allowing each provider to either evolve to it independently or choose not to.

Therefore, this paper moves forward the wider debate: "Is overprovisioning more likely to be cost-effective than rationing resources in a multi-service network?" [Odlyzko98, Breslau98]. Traditionally, the cost trade-off between the two has been between dumb simplicity and sophisticated complexity. By turning many traditional notions about charging on their head, we tip the balance towards resource rationing, but through sophisticated simplicity.

We argue that the charging mechanism should match the granularity of the provision of the network service - the packet. We argue that if the service granularity is finer than the charging granularity, price signals will never be able to optimise utilisation. Also commercial flexibility becomes limited. Systems that can charge on a per-packet basis can be specialised to charge for arbitrary aggregations of packets, but the reverse isn't so. Moreover, we show that very efficient per-packet charging is possible.

We suggest using the generally spare cycles on customer machines for measurement and aggregation (with their agreement). Clearly, this move to customer systems creates a security problem. We solve this using simple, random audit at a frequency tailored to each customer. The whole proposal is very much along the same lines as 'pay and display' parking policed by traffic wardens. Although we cannot yet estimate the cost of our proposals, we can confidently predict moving charging operations to customer machines will cost less to run than 'phone charging systems, despite charging per packet rather than per call.

However, the full benefits of a move to customer-based charging operations are dependent on another proposal in this paper: multicast dissemination of electronic tariffs and price changes to another proposal in this paper: multicast dissemination of electronic tariffs and price changes to cu: ners. This in turn has potentially far-reaching implications on the management, support and even marketing of network services. The greatest cost implication of this work could flow, not just from the dispersal of operational costs to customers, but from the potential ability to dynamically match network provision and demand using dynamic pricing.

Customer acceptance of dynamic pricing is a controversial subject. However we show that a network might still be managed by dynamic pricing even if many customers have a preference for price stability. We achieve this by suggesting that price stability could itself be offered at a price. We also question that preference for price stability will necessarily remain strong, given software could hide the nuisance of volatility. Even if the systems are only used to notify quarterly price changes in the short term, we believe it makes sense to build in the capability to announce far more volatile price changes for use in future years.

Although per-packet charging is often dismissed as impractical, per-packet policing is universally accepted, usually per-domain and sometimes per hop. We propose moving multi-service policing to the customer as well. Instead of the network denying access when a class of service becomes over-subscribed, we propose increasing the price to *avoid* congestion, but only as a last resort after borrowing resources [Floyd95]. Those customers that have most price sensitivity will then back off of their own accord. We assume, as dynamic pricing is introduced, a market will develop in software that controls adaptive applications dependent on price. The mechanisms proposed can cater for both long run and short lived price changes. They use zero bandwidth at the time of congestion, thus avoiding a congestion avalanche. This is achieved by downloading the algorithm relating price and congestion to customer machines. Thus, when the standard indications of congestion change, the relevant price movement can be calculated locally. If congestion pricing is found acceptable, policing can be removed from the data path, giving reduced latency and increased simplicity.

We must make it clear that this architecture *allows* for dynamic pricing and the removal of policing from the network, it doesn't require or mandate either of them. The advantage of a convenient way to quickly introduce new prices or pricing plans is sufficient rationale for the work.

We take an engineering approach, focussing on architecture, but also briefly reporting our implementation experience. Given the ideas are rather radical, this approach was chosen as a first step rather than modelling. We needed to understand what was required and clarify what was technically feasible. Ruth provides a review of other engineering approaches [Ruth97] with Edell *et al* [Edell95] and Brownlee [Brownlee94] being notable reports of large-scale practical experience. Much of the practical work gains some validity from the relatively new field of Internet economics, comprehensively collected at the 1995 MIT Internet Economics Workshop [Bailey95]. Utility curves mapping QoS classes to willingness to pay typically provide the bridge between economic theory and engineering practice [Shenker95].

In the next section, we outline the more difficult issues that have to be addressed in designing charging systems. We also try to disclose our own assumptions in order to give confidence that business models not possible within the present architecture are also not interesting. Note that we refrain from claiming to offer the pipe dream of a business model independent architecture. Section {3?} then presents the core of the paper - the principles of a good charging architecture, focussing only on issues of concern to a communications audience. We consider the fundamentals of how direction and transmission mode affect the flow of value from a network. This leads to a model of edge-pricing [Shenker96], but with asymmetric pricing, multi-service, multicast and aggregation added. We justify the choice of edge pricing over other models. We move on to consider the major building blocks of a general charging system, taking a fresh look at traditional practice in the field. We include inter-provider charging and the orthogonal issue of charging for the service the network offers to higher layers. For each aspect, we set down principles, describe systems we have invented to comply with the principles and then justify our choices. Section {4?} briefly describes the results of our work so far. The closing sections highlight the main weaknesses of the approach and suggest further work. Finally conclusions assess whether the high level requirements we are about to discuss have been met.

2. High level requirements

The overall aim is to provide all communications services over one network: data, real-time multimedia, network gaming etc. Our most difficult task is to neatly avoid (or at least address) the following perennial compromises, all, of course, at minimal cost:

- openness versus service differentiation
- security versus performance
- performance versus service flexibility

Instead of charging, over-provisioning, trust or coercion could be used to allocate detailed resources. A good charging architecture must inter-work with networks adopting these other approaches to be acceptable on the Internet, where management is highly federated.

It is well known that security requirements can lead to reduced performance. Therefore, wherever possible, we should use 'structural security' rather than cryptography or per access checks. An example of structural security is random audit, where customers are trusted to behave because of a broader deterrent.

Throughout the paper, we try to avoid creating the impression that the problem is difficult and complicated. Simplicity is the best way to allow future flexibility. Flexibility is also more likely if, for each aspect of the system, a model that is the superset of all other models is identified. However, generality does not imply abstraction - these general models must be translated into real mechanisms. Given the speed of change in the industry, flexibility is only possible if one thoroughly analyses and declares one's own assumptions.

2.1 Assumptions

We assume most customers have at least some price sensitivity. We intend to make no claims about this aspect (other than that the engineering is possible) until real customer trials give a measure of this factor. Odlyzko gives a comprehensive review of the literature on people's aversion to price fluctuation [Odlyzko97]. However, none of the studies separate people's aversion to risk from the nuisance of regular price changes, which software could alleviate.

To develop our model, we assume that a network is offered to customers as a 'black box' where end-systems trust the network to find the best route. The proposed architecture can charge for anything that packet headers can describe, so we believe we could charge for source routing if necessary. However, as source controlled routing isn't favoured, we ignore it to avoid unnecessarily complicated models. For similar reasons, we ignore inter-domain anycast. For multicast, which we use as the general case of unicast, the 'black-box' assumption means the network is trusted to find the best tree. In other words, we effectively assume a combination of two scenarios from Herzog *et al* [Herzog95] - 'sender pays' together with 'all receivers pay equally', but only equally on a per-domain basis. We don't assume no distance-related charging. However, if it is ever required, we assume distance can be approximated by source and destination addresses, rather than worrying about the length of the route between them.

As far as technological assumptions are concerned, we assume a packet network capable of some form of receiver initiated multicast. We assume *most* (not all) customer devices will be capable of running general purpose software and of having new software installed on a regular basis (by whatever means - flash ROM, mobile code, file download then installation etc), but we concentrate on the Java run-time system in our implementation.

We assume an ISP will be able to identify someone liable for the use of a network address (even if the actual user remains anonymous). This doesn't preclude anyone operating a service where they take responsibility for the actions of their customers without necessarily even knowing their identity. Similarly, given any Internet address, we assume it will always be possible for anyone on the Internet to quickly identify and contact the ISP that knows who is liable for that address. This in turn assumes that there will always be some means to form a contract with the liable party, no matter how implicitly. Consequently, we assume some local legal framework to make such contracts enforceable. We don't assume a common international legal framework. A common legal context is only needed for each customer-provider pair. One provider or one customer can span multiple contexts as long as the context of each of its contracts is clear. This ensures contracts remain enforceable at reasonable cost, a subtle factor missed in related work [Clark96, Fankhauser98].

Ou· sumed business model for an ISP has boundaries that match the service access points above and ᴗelow the network layer of the OSI stack. There will typically be different ISPs across the network layer of any end to end path. We only consider the case of a minimalist ISP - one that buys in lower and higher layer services (e.g. links or domain name service). This model still applies to ISPs that use their own links and services - the cost just becomes internalised. We assume networking intrinsically includes access, connectivity, routing, multicast and differentiated quality of packet forwarding.

However, the rate and burstiness at which service is delivered *from* the network layer is typically controlled with transport layer protocols. For instance TCP includes an implicit rate control 'contract' between network and end-system [RFC2001], while the RSVP flowspec is a very explicit contract [Zhang93]. We assume routine packet drop will not be an appropriate congestion control mechanism for many higher QoS classes - those where the delay of retransmit is to be avoided. This is one reason that various schemes have been proposed for selling the right to vary the rate control contract to the customer's advantage. MulTCP [Crowcroft98] is one proposal to pay to multiply the rates of increase allowed.

We assume that the near-universal case will be no more than one customer for each network address. However, the transport and higher layers *appear* to complicate the business model by multiplexing network service to multiple customers sharing the same network address. Examples are multi-user desktop machines and Web or video hosting services. But, the subcontracted customers only own identifiers buried deep within certain packets: respectively port numbers [Edell95] and URL fragments in our examples. So end-to-end protocols (including TCP) use addressing that doesn't key to any aspect of the network service. Therefore, apportioning network charges on this basis simply creates pressure for customers to implement new end-to-end protocols by private arrangement between themselves. This is why, pragmatically, a single party can be held responsible for all use of a network address, with any other users subcontracting this liability.

3. Principles and architecture

3.1 Definitions

In general we use terms for parts of charging systems as defined in Stiller *et al* [Stiller98] and its ETSI references. However, we reserve the term 'charging' to mean the whole process of pricing, accounting, applying pricing and payment, whereas Stiller *et al* use 'charging' to mean application of pricing to usage records, which we call 'application of pricing' or 'rating' for short. We use 'tariff' for the algorithm relating the class of service used to price.

We use terms for parts of the metering system as defined in Brownlee *et al* [RFC2063] - the Internet standards track real-time flow measurement (RTFM) architecture, which is broadly applicable to the present architecture. Because RTFM is intended for other uses than just charging, it uses the general term 'meter reader', whereas in our context we denote it more specifically as 'accounting'.

We use 'class of service' to mean a unique combination of network service mode (unicast, multicast etc.) and quality specification (latency, instantaneous bandwidth, reliability, jitter). The quality specification can either have fixed values for each parameter, or one class of service might fix some parameters while allowing the others to take a range of values specified by the customer.

3.2 Charging granularity

Most other work has started from the premise that per-packet charging is clearly impractical, suggesting instead aggregations such as quotas [Bohn93], addresses [Crowcroft96] or service level agreements (SLAs) [Clark95, Kelly97]. In consequence, network quality of service (QoS) architectures always embed some assumption about packet aggregation whether just for charging, or also for the QoS mechanism itself. For instance, the Internet Integrated Services Architecture (ISA) [RFC1633] uses a flow as the finest grained unit of QoS provisioning. Consequently this sets the minimum granularity when charging for QoS. On the other hand the Internet differentiated services (diff-serv) architecture proposes the packet as the unit of QoS provision, but suggests an SLA based on flows as the charging granularity. The assumption that the granularity of charging will have to be greater than the granularity of use then becomes embedded in the infrastructure (e.g. diff-serv pol ıg). So these approaches institutionalise a low utilisation factor, effectively assuming a business model for the network infrastructure, albeit quite a general one.

Further, if charging is on a coarser granularity than service provision, there will always be a gap between what is paid for and what is used. On the one hand, this creates an incentive for the customer to waste resources paid for but not needed (e.g. with crude robotic activity such as pre-fetching caches). On the other, it puts competitive pressure on the network provider to over-book capacity, thus eroding the original service guarantees. A diff-serv SLA suffers heavily from this flaw, especially an 'all-addresses' one.

Therefore we propose the charging mechanism should match the granularity of the network service - the packet. Not only because this doesn't artificially limit utilisation, but also because it allows much fuller commercial flexibility.

3.3 The direction of value

XXX analyses the factors affecting the value senders and receivers derive from communications [xxxa99]. It generalises edge-pricing for all transmission modes: unicast, multicast and aggregation (e.g. RSVP reservations). The conclusions are summarised here as any charging architecture must be based on the flow of value as well as cost. For those not familiar with the original edge-pricing model, we summarise it in two sentences first: A provider's edge-price covers its neighbours charges plus its own costs. Each provider is free to absorb variations in neighbour prices while setting its own prices.

XXX shows that the common case is for value to flow from the network provider outwards to each of its customers, whichever direction traffic is flowing. This is because the large majority of transmissions are with the consent of all ends. In consequence a general architecture must cater for a provider offering each class of service in each direction at a separate price. However, a large number of cases remain where all the ends may have a very different apportionment of the value of transmitting a packet compared to the send and receive prices charged by their local networking providers. If this discrepancy is large enough and prolonged enough, it should be rectified end-to-end, not through all the networking providers on the data path.

If all edge-provider income is normalised to local prices first, inter-provider charging then becomes simple wholesale of bulk traffic distinguished only by class, without worrying about how payment for each packet or flow was apportioned at the edge. This further allows the relationship between any two network providers to collapse to a single pair of prices per class of service for each direction of transmission. The price in either direction can be further thought of as the difference between the 'half' prices that each provider offers the other. Each provider's 'half' price can be considered to be for transmission between the edge in question and their respective remote edge of the Internet. XXX calls this model 'split edge-pricing'.

Using the end-to-end business models proposed in XXX allows this architecture to be efficient and scalable. The mechanisms for pricing, accounting and payment that we discuss below could be used in other models, such as [Clark96] or [Fankhauser98]. However, they lead to per flow charging in the core of the Internet and are to be avoided where possible.

3.4 Charging system topology

A usage-charging system must have two major functional parts if it is to control a system: a pricing control loop and metering to apportion charges. The question is, "Where should functions execute and where should data reside? - on routers, in packets, on the provider's operational support systems or on the customer's system(s)?"

We propose the principle that **the charging system should be as separate as possible from the transmission system**. Information in the network should only determine the behaviour of machines forwarding it. Any charges for this behaviour should be referred to indirectly, by mapping between what packet fields do and how much will be charged for doing it - in effect a contract or quotation. This gives a 'zero bits for charging' solution in contrast to one or two bit solutions [Mav97, Nichols97].

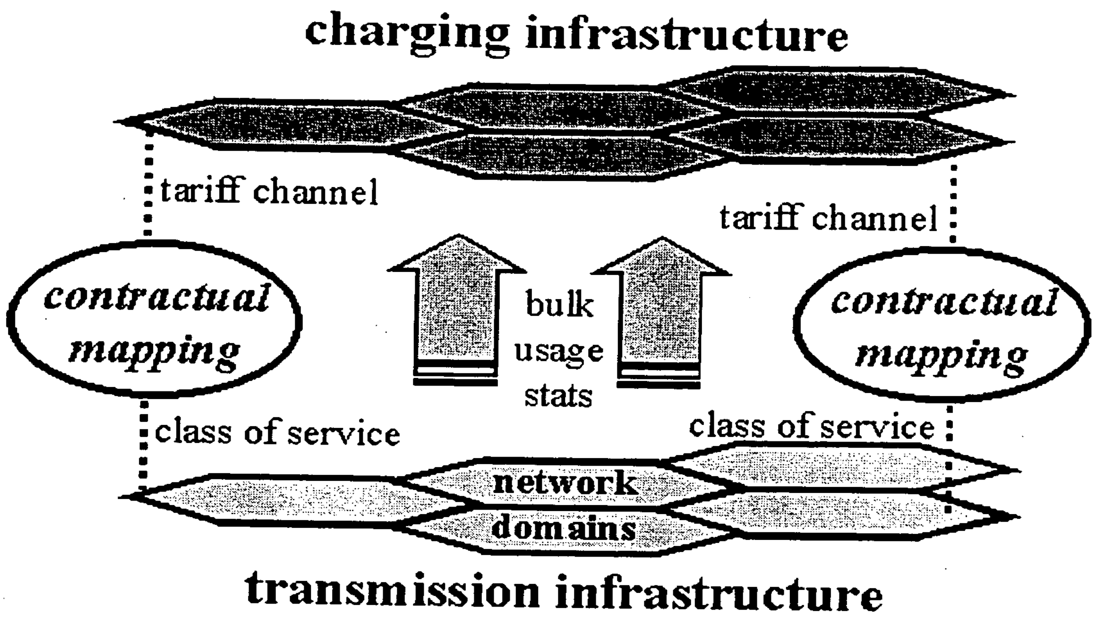


Fig {3.1?} - Zero bits for charging

Fig {3.1?} shows different contracts for different domains mapping behaviours in the transmission infrastructure to tariffs in the charging infrastructure. Mapping between the two infrastructures via dynamic contracts allows each to evolve independently. Because each domain can offer its own contracts mapping its own traffic classes to its own tariffs this gives each provider commercial freedom. It also allows future charging schemes to be arbitrarily complex.

We also propose the principle of **moving as much as possible to the customer machine**. This helps scalability, improves customer control and improves responsiveness to price. We propose that the above contracts, or at least the tariffs they contain, should be code running on customer machines - active tariff objects. This brings the customer machine fully into the pricing control loop. All the inputs to any typical tariff algorithm are already available on the customer system, or can easily be made available. This gives the customer immediate charging feedback so that her behaviour can be optimised to the price signals of the provider. Over time, we expect that the customer will suffer less and less interaction in order to accept or reject prices, as software agents are written to control application behaviour based on customer spending policies, application preferences and these tariff algorithms.

More radically, we also propose to move metering and accounting to customer machines. The cost of meters placed at edge-routers will grow proportionally to the bandwidth that more and more end systems will throw at them. Worse, the charging systems behind the meters tend to be more disk speed limited than, say, multimedia systems, because the former demand data reliability and audit. Worse still, in a multi-service network, metering not only volume but also service rate and burstiness will be part of the requirements. This is a particularly heavy load for routers to bear both in terms of memory and processor load, as experience with MulTCP and RSVP has shown. Even though very efficient token bucket mechanisms can be used that have only a tiny incremental effect per packet, the aggregate effect on a router handling many flows is still highly significant. For this and other reasons, an applicability statement had to be made in 1997 limiting RSVP to small-scale scope for the foreseeable future [RFC2208].

On the other hand, to measure the packet activity of one host using the same host is naturally scalable. The most likely party interested in detailed usage data is the customer or local software managing resources on her behalf. The provider is generally satisfied with a statistical view. Therefore it makes sense to aggregate all the data at the customer before forwarding it to the provider. As long as either party can tailor the level of detail at any one time, it is best for the default to cater for the common case.

Thus, each customer will be able to calculate her own bill continuously and pay it on whatever schedule might have been previously agreed - self-billing. However, even though the support systems will be running remotely, the provider will want to retain the primary levers of control:

- pricing levels
- data reporting frequency and granularity
- audit of the revenue flow

The provider retains control over price levels by invoking a remote method on the tariff objects on customer machines. Below, we briefly describe how the provider can also update the whole tariff on the fly using the same mechanism. For efficiency, these tariff changes can be disseminated by multicast. The customer would have to permanently join the relevant multicast groups to hear tariff changes for her class of service. The contract would need to make this clear - the contract could even include code to ensure this happened.

The provider can retain control over data reporting from the accounting system by remotely accessing the customer machine. The contract would need to make access a condition of service. Instead of the quarterly or monthly batch bills of today, these billing reports might be requested to be triggered every day, every hour or even every few minutes. The provider simply has to issue new instructions to change the frequency of reports. The same mechanisms can be used to ensure all future reports give data of a different granularity. This might be required to support a new marketing campaign, for instance. Multicast could be used unless per customer changes were required. It would also be possible for the provider to configure the reporting system back to the traditional billing mode with metering done by the provider and bills being sent to the customer. This might be necessary if there were a restricted back channel (e.g. satellite).

To audit the revenue flow, contradicting our previous statement, we propose the provider *will* run metering on edge routers, but only on a sampled basis. At any time, the provider might choose to sample a particular customer without her knowledge. Measurements would be collected and aggregated to see if they matched the regular reports being sent back from the customer accounting system. The sample would have to be taken for a whole multiple of the customer's reporting periods, and in phase lock. The two would be compared and if they were within tolerance, the provider measurements would be discarded. There might be a few customers being sampled per edge router at any one time, but the measurement load would never approach that required for full metering on the edge router. It would be important to ensure no single customer could detect that her data was being sampled (e.g. by a marginally increased end-to-end latency).

If there were a discrepancy between the customer's and provider's accounts, the provider would store both sets of data as evidence. The sampling rate might be increased on that customer. If the discrepancy persisted for future samples, eventually some penalty action would be triggered. For instance, customers might have been asked to pay a deposit and the penalty might be to take the shortfall from the deposit. For certain customers, it might be necessary to just run the system in the traditional billing mode - effectively 100% sampling.

3.5 Pricing

3.5.1 Tariff dissemination

We have suggested that tariffs could be 'announced' over multicast channels, much as sessions are described with the session description protocol (SDP) [RFC2327] then announced on the mbone. Tariff announcements would be regularly repeated on the soft state model. We use soft state to give timeliness to new announcements without polling and to add the reliability of repeated sends to unreliable Internet multicast. Unlike SDP, which consists of text fields, we wish a tariff to be executable code for full flexibility. In our testbed, we serialise Java bytecode, authenticate it, add a serial number then multicast it. However, there is no reason why a text-based protocol couldn't be invented that would be parsed on the local machine in order to change local tariff parameters.

There is more than a passing similarity between session descriptions and tariffs. Both have a start time and end time (which may be unspecified) and a time when they were first announced. Both might need authentication. Also we suggest that at least two multicast channels should be used per provider. One to disseminate the tariff classes themselves while the other would announce metadata about the tariffs (e.g. the current version number, a description, regular price level fluctuations etc.). With edge-pricing, customers would only have to listen to their local provider's channels. A provider may also make tariffs available by request-reply (e.g. Web) for newly booting machines. The tariff announcement period could vary slowly. Each announcement would have to state the maximum period for at least the next few announcements so that customer machines would quickly be aware of missed announcements.

It is clear that there is a potential problem if the customer system doesn't hold an up to date tariff. However, the provider cannot allow customers to use a lower rate if they claim they haven't heard a more recent price rise announcement. This would be an obvious loophole in security. Customers willing to risk continuing to use a service without knowing the exact price would have to include the version number of the tariff they were using when reporting their accounts to the provider, and the fact they were aware it might be out of date. If they subsequently found they had missed a more recent version, they would be able to send an adjustment. If price changes showed a history of being minimal, it is likely most customers would be willing to take such risks - the discrepancy would be a small proportion of a small amount.

There is clearly an issue of security from the customers' point of view, if they are expected to allow their provider to send them arbitrary code that will be executed on their own machine on the fly. There are two issues: "Will the code do other things besides change tariffs?" and, even if it does only change tariffs, "Are the new tariffs acceptable?" The Results and Further Work sections briefly discuss these issues.

There is also an issue of customer acceptance of dynamic pricing. However, first, we must clarify that we are not saying these mechanisms have to be used to update prices without any notice. A new tariff or rate has a specific field that states when it becomes active. Therefore, the system can emulate the fairly slow time-scales providers allow today for customers to consider future price changes. However, it is also important that a general architecture can cater for a future that may be much more dynamic. Thus, the 'tariff activation time' field can be set to null which implies the new tariff is activated immediately on receipt. The contract might assure the customer that no price updates would be imposed without a certain period of notice, giving legal redress against fraudulent providers. We have proposed the multicast announcement model so that it can cope with a possible future where prices are changing very rapidly. A half round-trip time is the physical minimum delay possible in the price control loop, which multicast achieves.

We also propose that it is perfectly possible to include a price for price stability itself in a tariff. Risk averse customers could choose to pay a premium for price stability, while those willing to play the market could take the spot price at any one time in the expectation that their average costs would be less. We propose that one simple algorithm linking a percentage premium to a period of stability could be applied to the spot tariffs of all classes of service. Disseminating this as an active tariff would offer the customer a variable degree of price stability. Of course, the provider might want to vary the price of price stability - but we assume they would have to give notice for that! Any customer wishing to use a more stable price would have to declare their intentions up front. She would have to state which address range she needed a stable price for, and for how long, in a message to the provider. Otherwise, because she is billing herself, she could decide in retrospect whether the stable price or the spot price had been more advantageous. Other types of 'futures market' would also be possible. Customers might make this decision on a long-term basis for all their usage, or just for specific sessions. An example of the latter might be a long conference where the local customer didn't want to commit to starting without knowing the exact cost. Further work is required to establish whether there will be enough room between service tariffs to fit a price for price stability without distorting the distinctions between the prices of different service classes.

It appears that the provider has to have a certain degree of trust in the customer's idea of time (and vice versa). However, the random audit technique deters any deliberate clock shifting by customers, because they can only fool their own system, not the provider's. The regular tariff announcements provide a convenient channel to state the provider's view of the time while the accounting reports provide a return channel where the customer can state the time they received the tariff. Thus, the round trip time is constantly being agreed between the two ends. There is scope for customers to undetectably shift their clocks to their advantage within the deviation of the round trip time, but their financial gain from such shifts would be minimal.

3.5.2 Price control

So far, we have said little about how a provider would determine price levels. This is really outside the scope of the current paper, but a few words will put the subsequent discussion on congestion pricing in context. We suggest the following steps before any price rise is considered:

1. the network re-routing around congestion
2. the network borrowing capacity from "lower" classes of service
3. the network introducing or at least requesting extra capacity (possibly automatically)
4. the network provider (possibly through a rule-based system) deciding a price change is commercially advantageous

We are not suggesting that the provider should *always* avoid congestion using price. It may be that more revenue can be gained by allowing the network to fill and denying service to subsequent requests. We are merely pointing out that we can use the hierarchy of service classes to advantage because they can borrow capacity from each other. We can manage the load on each logical service class with price because we can always grow the size of each logical class (as long as there is sufficient best effort buffer at the bottom). It is only the best effort level that needs to have the ability to deny service - and it already does that by dropping packets.

It is also possible that each tariff for each class of service for each direction could be further subdivided by various ranges of local and remote address. Thus, if there were spare capacity in just one region of a network, it might be possible to offer lower prices to customers based on the combination of their address and the remote address of any flow. These combinations would be the ones known to route through the lightly loaded region. We do not distance based pricing is very likely, but it is possible within the present architecture. Tariffs can include weightings to be applied to the standard price for combinations of source and destination address prefix lists.

The other side of the price control loop is just as important to understand - the customer reaction to price. We assume software components similar to that in Tassel *et al* [TasselBri97] will be added to many types of adaptive applications, but the study of how users will configure their price behaviour is left for future work.

3.5.3 Congestion avoidance pricing

Above, we suggest how to control price either for whole domains or, if necessary, for regions within a domain. However, our mechanism is not scalable enough to control the congestion of any queue on any interface on any router in the Internet. Because data is bursty, such congestion can appear and disappear in different queues fairly rapidly. This would require a price announcement from every queue and some way of targeting these announcements just to those edge-systems transmitting through that queue at that time. Worse we really want to charge for ignoring congestion back-off protocols, rather than charge for congestion itself. Even if this could be done, we wish to allow each provider commercial freedom. This would include the freedom to absorb peaks and troughs in prices from neighbouring providers in order to present simpler pricing to neighbouring customers. Therefore, congestion prices would become lost in the noise.

Instead, the mechanism we suggest is to use existing congestion signalling, which effectively creates a channel from any point of congestion to the end-systems using it. Various schemes exist or are proposed for the Internet, which avoid a congestion avalanche by highlighting the packets that are causing the congestion. This can either be done implicitly by dropping them (as in TCP and RTCP [RFC1889]) or explicitly, e.g. using the explicit congestion notification (ECN) bits [Floyd94] proposed for diff-serv. All these methods require the co-operation of the receiver to notify the sender of congestion through an end-to-end protocol.

We *could* require all these congestion control protocols to be changed to make the marks on the packets represent a shadow price [Kelly98]. Instead, we can keep the protocols as they are and re-introduce the commercial freedom given by edge-pricing. We do this by including congestion signalling as one of the parameters driving the tariff on the end-system. That is, by assuming a competitive market, we can allow providers to each map this shadow price to their own real market price. Thus, congestion can still be signalled in a standard way, but it can refer to charging inf nation indirectly. This is again in compliance with the principle of keeping transmission and cha ging infrastructures separate. Backbone providers would charge their neighbours heavy penalti for congestion being ignored. In turn their neighbours would translate these into heavy penalties for end customers if they ignored congestion.

We propose this for congestion avoidance rather than congestion control. We should be clear that we are talking about congestion of a logical class of service. This scheme is unlikely to make sense for best effort congestion. Our proposal would allow a logical service class that is approaching capacity to start penalising end-systems that didn't back off at the first sign of congestion. This would be particularly appropriate where a logical service class was designed for applications that couldn't tolerate much (or any) packet drop - the reason ECN was proposed in the first place. As already said, we assume resource would already have been borrowed to the limit from other classes of service.

Both explicit or implicit congestion signalling can be measured and therefore form the basis of charging at the end-system or audit at the edge-router. In both cases, it may be necessary to bury into the transport layer header of the returning packets to measure whether the receiver is notifying the sender of congestion. It may also be necessary for the ISP at the other end to check in a similar way that receivers are complying with the end-to-end congestion protocol. Although this process would be expensive to the router, it would only be on a small sample basis. The following steps would have to be added to the above checklist, before any host would ignore congestion back-off requirements and instead pay the price:

1. the end-system should consider setting QoS requirements to a "higher" class (if cheaper than the fine for ignoring congestion at the current class)
2. the end-system should decide it is essential to ignore the congestion, given the fine for doing so might be quite high
3. both (all) end-systems should agree to ignore the congestion

Kelly has already shown congestion pricing can be stable if everyone is using the same algorithm. However, making congestion control dependent on such unpredictable factors as customer sensitivity to real market prices and provider profit motive could be dangerous for total network stability. More work is required to find if there is a class of algorithms that simultaneously allow commercial freedom but still exhibit stability in reasonable scenarios.

3.5.4 Optimistic access control

Typically, three per-packet operations are added to routing and forwarding to create a multi-service packet network: classification, policing and scheduling. Of these, classification and policing involve look-ups into tables of potentially dynamic state. These tables can become large, requiring a few levels of look-up hierarchy, each of which takes resource. Further, state introduces complexity in keeping it current, keeping it in the right place and garbage collecting the resources it uses when it is finished with. Soft state is often used to alleviate these problems. However, as routes change, keeping route dependent state on the correct router introduces further complexity.

As far as classification is concerned, diff-serv has already taken the step of confining the number of possible classifications to a small non-dynamic table, thereby making classification and scheduling simple and independent of flow state. However, in all known multi-service network proposals, policing always involves blocking the flow whilst checking every packet against restrictions - 'pessimistic access control' [Linington98]. Examples are token bucket policing against RSVP flowspecs or diff-serv SLAs [Clark95]. The flow cannot continue in parallel to the access control process because the only punishment available is to hurt the packet in question. This is a particular problem in connectionless networks where any one packet cannot necessarily be related to a later packet.

Instead we propose 'optimistic access control' where packets are allowed through without checking, but customers know they will have to answer for whatever they do. The optimistic model can only exist within some wider pessimistic context. A charging system offers just such a context. For instance, a customer may only be given an Internet account after revealing her verifiable identity, or paying a deposit. Once past this pessimistic hurdle, optimistic access to more specific parts of the service can be allowed, protected only by a punishment if payment doesn't match metered activity (respectively legal action or docking the deposit). Because metering can proceed on the memory cor ɿf the header, in parallel to the flow of the service, latency is always kept low in this optimistic mou.ı. Thus, the combination of an optimistic within a pessimistic model needs just a single blocking test. The effect of this single test can persist for months or even years.

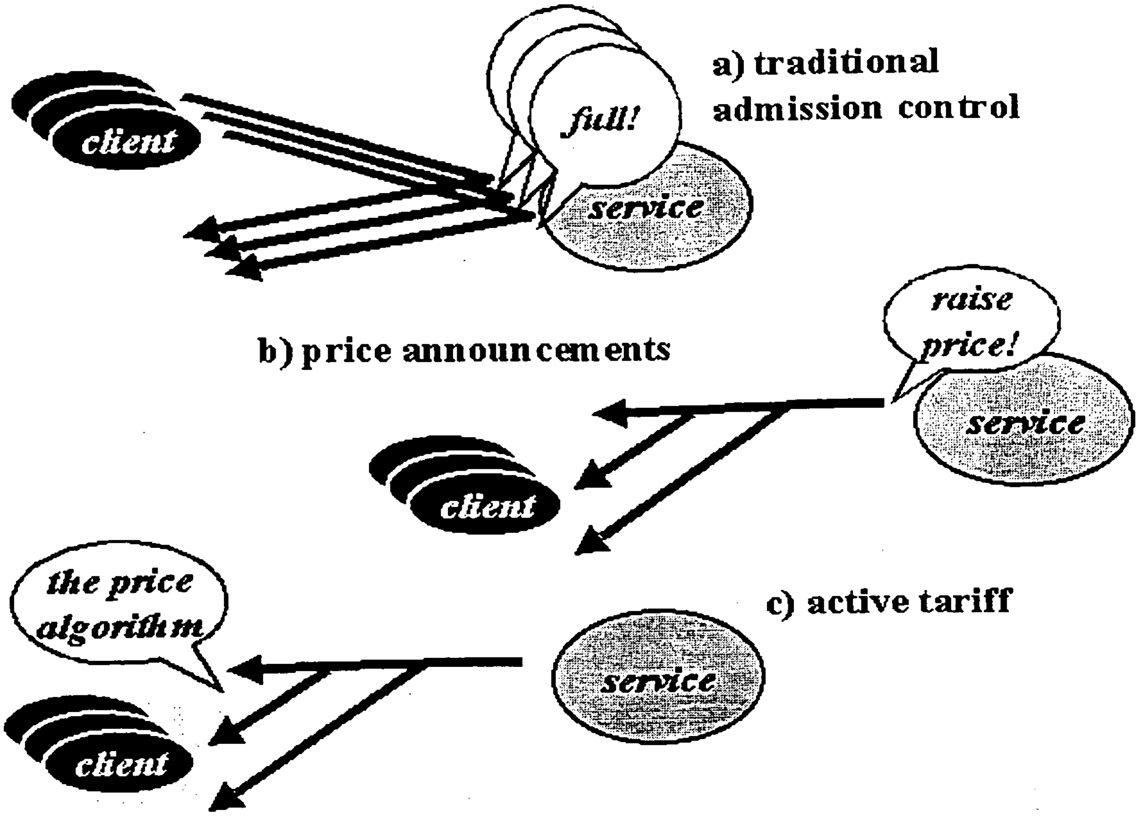


Fig {3.2?} - Admission control at source

Through adding a dynamic price mechanism, we can ensure that those people least willing to use a service at a certain price *deny themselves* access. The provider is then freed from having to fend off flash crowds with an avalanche of admission control signalling (Fig {3.2?}) or the complexity of RSVP blockade state. With the tariff pre-installed on the customer machine, we can even avoid any 'raise price' signalling at the time of congestion. Thus, potentially, we have even moved both admission control and policing to the customer machine.

3.6 Metering, accounting and payment

We now describe a system architecture (Fig {3.3?}) for measuring usage and paying for it, confining ourselves to issues relevant to a communications audience. The left to right division is between customer and provider systems, with otherwise identical classes of objects distinguished respectively by the subscripts 'c' or 'p'. We have already described how the primary charging system can be operated by either the provider or the customer, depending on whether the system is configured for traditional or self-billing. This explains the essential symmetry of the figure. The networking service is shown flowing from provider to customer, regardless of transmission direction. The main chains of objects on either side drive a flow of data upwards:

- service usage is metered, M, which also aggregates to simple rules arbitrated by the meter controller, MC
- accounting, Act, consolidates results from possibly a number of meters and deals with hard storage
- rating, Ra, is where prices are applied to the usage data to calculate charges.
- ultimately payment, Pa, of the calculated charges may be required

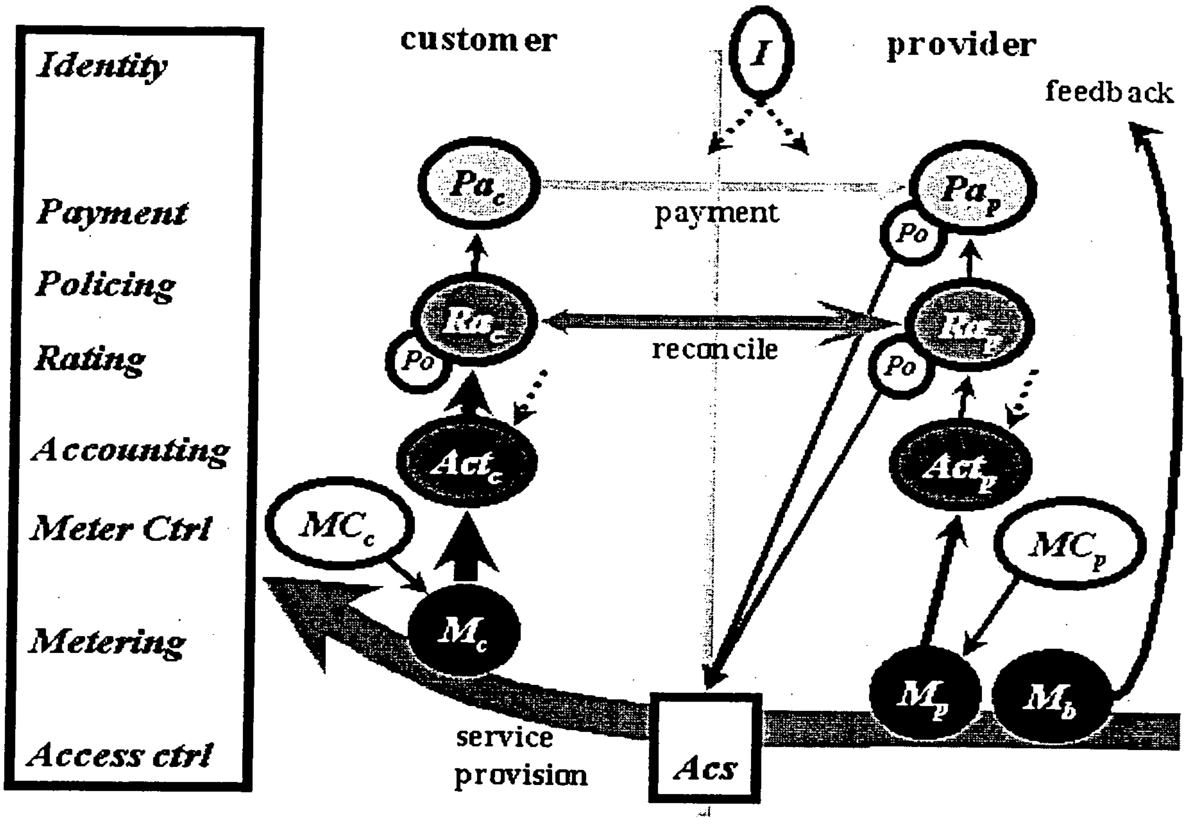


Fig {3.3?} - Metering, accounting, rating and payment architecture

External control (not shown) of each side of these systems is primarily by the contract, particularly the tariff, which determines what to measure, what and how often to report, and current pricing. An agent of the customer or provider respectively arbitrates this control. Control is typically implemented by caching a policy in the relevant object, whilst listening for policy update events. Feedback to complete the control loop is shown flowing from bulk measurement on the provider side ('bulk' means queue lengths etc. rather than individual flows).

We use the term reconciliation to cover the 'pay-and-display' model and traditional billing. In both cases, the aim is agreement over the level of usage. The most efficient level to reconcile customer and provider is after prices have been applied, as shown. Both sides may wish to police reconciliation, with failure to reconcile triggering some defensive action. In the case of the provider this is shown as ceasing access, but less drastic measures may precede this. Payment is also policed by the provider - it is no use agreeing on the amount owing then not checking the amount paid. Rather than expect total accuracy in reconciliation, the system would be designed to ensure discrepancies due to system problems were as likely to be positive as negative. Arbitrary packet loss from flows in either direction between the two meters clearly makes this impossible. Therefore the two meters should ideally sit on either end of the same link. If link layer error correction is in use, metering should occur after correction. Metering should be as low as possible in the stack to avoid any contention for resources within the stack due to packets with different QoS classes. Even if best effort traffic were being discarded, if a link layer QoS scheme were used (e.g. [IEEE802.1p]), metering higher QoS traffic would at least show no discrepancies. Ultimately, as long as packet losses were low, both parties could write them off by widening their tolerance to reconciliation discrepancies. This introduces a security flaw, as either side can try to probe the limits of the other's tolerance. Marginally higher prices for everyone would have to cover these potential losses - similar to the cost of pilfering, spillage and spoilage and in the retail trades.

Next we discuss *who* is considered liable for charges. We assert that network metering should only be concerned with apportioning use of the network service to various network addresses. The identity service, I, in the figure is responsible for the mapping between network address and real world identities. This is another example of the principle of separation between network and ch ing. Address allocations themselves need to be 'metered' and 'usage data' sent to the accounting function to be consolidated with the network usage-data for each address (shown as dotted arrows). Address allocation schemes must be careful to allow time between de-allocating an address and re-allocating it to someone else. All packets containing this address must have cleared the network and all the dependent systems must then have had time to flush their caches of the mapping. Examples of address allocation schemes are dynamic host configuration protocol (DCHP), network address translators (NATs), mobile cell hand-off and multicast address allocations, joins and leaves. A many to one mapping between addresses and identities must be assumed. For instance, one customer becomes liable for reception charges for any traffic to a multicast address group as soon as the join to the group from her machine is measured. Time must also be allowed after allocating an address for access control systems to check their policy concerning the new identity.

Having established a mapping between a network address and an identity, liability for charges depends on the contractual position. It may be that the local customer is only liable if one of its addresses is in the source field of the packet, or the destination or both. Whatever, from an architectural point of view, all that is necessary is to ensure accounting and payment are totally separate. The process of reconciling the account with the provider proceeds in two stages: first it is agreed which usage occurred, then which usage the customer is liable to pay for. That is, the local customer's account should always include all the usage data for both sent and received traffic, whoever might pay for it. Account reconciliation can occur on a completely different schedule to payment.

Because the architecture we have described applies all the principles given, it will charge correctly even in the most demanding scenario. However, in inter-domain scenarios it will only be efficient by using the 'split edge-pricing' model. For instance, XXX works through an example scenario with inter-domain multicast and heterogeneous QoS per receiver [xxxa99]. That example also shows how different apportionments of charges between senders and receivers can be achieved for multicast, again by dealing separately with such issues - end-to-end.

3.7 Charging inter-operation

3.7.1 Higher layer systems

In our discussions above, and those on the direction of value, we showed how the remote end of a transmission might want to 'wholesale' some charges from the local ISP then 'retail' them back to the user at the local end. Examples are: a video on demand service that 'bundles' transmission quality charges with their videos; or the 'originator pays' model used in traditional telephony. In general, any third party not even involved in the communication might pay the local network charges. Examples are the user's employer or an agency organising a videoconference.

The simple step taken above of separating accounting from payment, combined with multicast pricing announcements, allows arbitrarily complex business models to be built on the foundations of the present architecture. All the relevant information is available for verification by third parties, whatever degree of trust they put in the network users for whom they are paying. For instance, prices are authenticated by the ISP. Even if the ISP is coy about allowing non-customers to join the pricing multicast, the local user can unicast the authenticated tariff object to the third party. If requested by the third party, the relevant usage data can also be authenticated and sent, either by the ISP or the local user. We are not saying these arrangements *have* to be authenticated - the third party might simply offer to cover charges up to a flat ceiling, whatever they actually were.

3.7.2 Inter-provider

So far, it may have been assumed that all scenarios only applied to edge providers and customers. However, only the previous section, on inter-operation with higher layers, is specifically about end systems. The rest of the architecture is just as applicable to an inter-provider relationship. XXX shows how any two ISPs have a customer-provider relationship, depending only on the sign of the difference between the 'half' prices that they charge each other. We propose that all the principles and mechanisms described above apply equally to this relationship - the architecture is recursive.

For pricing, tariffs could be announced from each provider to its neighbours. Each provider would base its prices on the costs it was receiving from its neighbours, combined with its own commercial ob' tives. Even if the edge-pricing model weren't used, non-neighbouring providers could still rec.ve these price announcements.

For accounting reconciliation, each pair of neighbours could either both measure or, for efficiency, agree only one need measure - then the other need only sample. In fact this is similar to the standardised model for international carrier accounting in the PSTN [ITU D.150]. Here, both parties measure everything but the payment is based on the *payer's* data. Unlike the PSTN though, as long as payments are normalised end-to-end first (as recommended earlier) the choice of factors to measure at any inter-provider boundary is local to that pair of providers. Thus, even if edge-provider to edge-customer charging is per packet, charging between the same edge-provider and a peer-provider could be by average link utilisation per service class. The obvious restriction to this heterogeneity is the need for some causal relation between one measurement system and another. Otherwise there would be no way to relate costs to prices and no price back-pressure.

Recursion also applies for corporate customers, where there may be a need to apportion costs between departments. The accounting reports would first flow between end-systems and departmental accounting systems. Then departments would aggregate and forward on to both the corporate and provider systems. The model can even be applied recursively to multi-user machines. Each user simply runs an accounting object that reports to the machine's accounting object. Intra-machine accounting would be based on different addressing as already discussed under assumptions. In all these recursive models, the detailed accounting within one system can either be encapsulated from the broader system or revealed if the commercial relationship requires it. If there is any shortfall between the total of all the details and the total the provider expects, that is the responsibility of the party at that level to underwrite.

This neatly brings us to the final issue to resolve in this paper: "What about failure to deliver the specified service?" With an end-to-end service, it can be very costly to apportion blame when things go awry. If the customer has paid for reliability, who should give the refund if a packet is dropped? If latency is guaranteed, who gives the refund if some network has used too much of the delay budget? We propose a pragmatic principle for these circumstances. If a customer disputes payment and their local provider accepts their case (or can't be bothered to argue) all providers on the end-to-end path share the same fate, losing the associated revenue. This is akin to peering between ISPs today, but need only be applied to the exceptional cases of failure. Hence we call this 'exception peering'.

4. Early results

Other than producing the architecture summarised here, we have also implemented research prototypes of the primary system components described above, primarily in Java. These include the provider and customer ends of the tariff dissemination mechanism and a generic accounting system for either providers or customers with remote control capabilities. We implement active tariffs in Java, the most complex one to date being 8kB on the wire. The receiver end includes a modified class loader to switch to new tariff objects on the fly. Remote method calls are turned into objects for dissemination then back into calls once received. The incoming object currently undergoes rudimentary checks: for a trusted code signature and to check conformance with the expected interface by introspecting the class of the object. We have also built a component for integrating local charging into any Java application. It requires only a minor edit to the relevant socket calls, being based on Tassel *et al* [TasselBri97].

The whole charging system currently consumes 1.6MB of storage on the customer device, including 875kB for a stripped NeTraMeT [Brownlee97], which we use for some metering (but not for RSVP). We are about to start more scientific experiments, but initial indications show that the charging system consumes about 3% of CPU time on a 400MHz Pentium II for moderate traffic loads. Because the code is intended as a flexible research testbed, no attempt at streamlining has been made. This might reduce storage and system load further.

5. Limitations and further work

The ideas presented here are rather radical and hence prone to unpredicted weaknesses appearing as they are developed. Currently there are six areas that cause us most concern:

- Dispersal of software critical to an ISP's revenue flow into an unreliable installation environment. Although we can fall back to traditional billing, our scalability benefits rely on this fall back not being needed too often. This should be less of a problem with single purpose devices than general-purpose computers.
- Customer acceptance of foreign code installing and running itself on their systems. We can make rudimentary attempts to address this, but we also rely on solutions emerging from the mobile agents and active networking fields.
- Resource allocation purely by price could lead to bursty hogging. The laws of economics should protect us against the static effects of hogging [Clark95] - the hogs will pay for the capacity they hog, leaving the remainder for the rest of us. However, heavily bursty traffic from relatively few price insensitive customers may cause a disproportionate need for over-capacity.
- Demand management by price could fail if deterrents like penalties or credit blacklisting suddenly became too weak, e.g. due to crisis events. Rather than remove admission control from the network, it may be advisory to leave a vestigial 'defence in depth' to be turned on in emergencies.
- Meter discrepancies between customer and provider. This becomes a problem with lossy access link technologies.
- Lack of experience of dynamic pricing behaviour of customers and providers. Experiments are needed where risk aversity is distinguished from the nuisance of dynamic pricing.

All these areas except the first are subject to further research. This paper has been presented before we have any macro scale predictions from scientific experiment, modelling or simulation, on the premise that the ideas are of interest in their own right.

6. Conclusions

A number of innovations have been proposed which will need further modelling and testing. Nonetheless, it can justifiably be claimed that this paper concentrates on giving answers rather than raising issues; on laying down simple principles rather than creating complexity. It can be concluded that these rather radical proposals have a strong chance of producing a highly scalable, lightweight, high performance, secure and open system that will still allow business flexibility, but above all, will be cheap to run.

A single price for each direction of each class of service at each boundary between networks is recommended. This is termed 'split edge-pricing'. It is argued that this will be necessary and sufficient to cater for a multicast multi-service network, while still giving each provider commercial freedom. The solution is intrinsically designed to inter-work with other approaches allowing each provider to either evolve to it independently or choose not to. Allowing anyone to settle anyone else's account by end-to-end arrangement ensures the common case apportionment of charges between senders and receivers is optimised, while giving flexibility for exceptions. A similar solution is suggested for apportioning the blame for failure to meet end-to-end service obligations. Assuming such circumstances will be exceptional, rather than spending resources establishing blame, it is recommended that all providers on the path should share the loss. This is termed 'exception peering'.

A radical shift of charging functions from provider to customer systems is proposed. The provider maintains security by simple random audit. The provider maintains control by distributing active tariff objects to all customers and by keeping the contractual right to remote control of data reporting and price levels. A multicast-based dissemination technique is proposed to achieve this remote control efficiently. The customer, on the other hand, gains full control of all other aspects of the charging system. All the provider's price signals are available locally and can be used by the customer or her software to aid decision-making. Timely feedback on the financial implications of every action are also available locally. Current networks totally rely on customers not fully utilising the services being sold. As customer software becomes increasingly sophisticated this assumption could become dangerous. Instead it is proposed that customers should be given exact price signals so they can co-operate, rather than compete, with the provider's goals.

The proposed charging architecture allows as many factors as possible to be configurable. Provider cor ·l over price and reporting are the only assumptions about who might trust whom to do what. Gra..ularity of all service charging can be as low as a single packet, but per-flow or inter-provider bulk aggregations are just as possible. However, the processing load of aggregation is distributed across customer machines. The architecture should also cater for 'clock speed' improvements from the quarterly billing and price change cycles of today to sub-second reports and price changes in the future if required. Suggesting that price stability can be offered at a price. it is shown how risk averse customers can be offered stable pricing while others can pay generally lower but more volatile prices.

It is argued that flow policing would be unnecessary in any domains where this charging architecture was in place. Instead, the customers could be relied on to police themselves due to price back-pressure and wider deterrents against defaulting on payments. Thus. effectively, flow-based access control can also be moved to end-systems. Instead of policing every packet at every router. or at least at every border router, this offers the alternative of merely measuring every packet once at each end of its transmission. Further, measuring can be done in parallel to forwarding, whereas policing, although very lightweight, requires forwarding to be blocked until it completes. Combined with simple classification schemes like those proposed for Internet diff-serv, this would remove any need for flow related state on routers. This would also remove the complexity required to keep such state on the right routers if routes needed to change. The proposed model is termed 'optimistic access control'.

The authors are aware that these proposals are as yet untested. Weaknesses that give concern have been listed. Subsequent discussion about this and other equally valid approaches to the same problem will doubtless highlight further limitations. However, the authors believe it is imperative that networks should remain simple. This paper offers a multi-service packet network with the complexity of policing and charging completely separated out except at the ends. This leaves the network infrastructure clear to simply classify, route, schedule and forward.

7. Acknowledgements

{Removed for anonymisation}

8. References

[Bailey95] J. Bailey, S. Gillett, D. Gingold, B. Leida, D. Melcher, J. Reagle, J. Roh, R. Rothstein, and G. Seale, "Internet Economics Workshop Notes," Research Program on Communications Policy, MIT, Mar 1995. <URL:http://www.press.umich.edu/jep/works/BailWNotes.html>

[Bohn93] Roger Bohn(Uni of CA-San Diego), Hans-Werner Braun & Kimberly C Claffy (San Diego Supercomputer Center) and Stephen Wolff (NSF DNCRI), "Mitigating the coming Internet crunch: multiple service levels via Precedence", Technical Report, University of California-San Diego, San Diego Supercomputer Center and NSF, Nov 1993, <URL:http://www.nlanr.net/Papers/mcic.html>

[Breslau98] Lee Breslau and Scott Shenker (Xerox PARC), "Best-Effort versus Reservations: A Simple Comparative Analysis", in Proceedings of ACM/SIGCOMM '98, Vancouver, Sep. 1998, <URL:http://www.acm.org/sigcomm/sigcomm98/tp/abs_01.html>

[Brownlee94] Nevil J. Brownlee (Uni. of Auckland), "New Zealand experiences with network traffic charging", usage-based access charging, ConneXions Volume 8, No. 12, Dec 1994, <URL:http://www.press.umich.edu/jep/works/BrownNewZe.html>

[Brownlee97] Nevil J. Brownlee (Uni. of Auckland), "The NeTraMet System", Software Release Notes, Dec 1997, <URL:http://www.auckland.ac.nz/net/Accounting/ntm.Release.note.html>

[Clark95] David D Clark (MIT), A Model for Cost Allocation and Pricing in the Internet, presented at MIT Workshop on Internet Economics, Mar 1995, <URL:http://www.press.umich.edu/jep/works/ClarkModel.html>

[Clark96] David D Clark (MIT), "Combining Sender and Receiver Payments in the Internet", in Interconnection and the Internet. Edited by G. Rosston and D. Waterman, Oct 1996, <URL:http://diffserv.lcs.mit.edu/>

[Crowcroft96] Jon Crowcroft (UCL), "Pricing the Internet", in IEE Colloquium on Charging for ATM (ref. no. 96/222) 12 Nov 1996, pp1/1-4.

[Cr..croft98] Jon Crowcroft & Philippe Occhslin (UCL), "Differentiated End to End Internet Services using a Weighted Proportional Fair Sharing TCP", Apr 1998, <URL:ftp://cs.ucl.ac.uk/darpa/multcp.ps> or <URL: http://www.cs.ucl.ac.uk/staff/J.Crowcroft/hipparch/pricing.html>

[Edell95] Richard Edell (Uni of CA-Berkeley), Nick McKeown (Stanford), and Pravin Varaiya (Uni of CA-Berkeley), "Billing Users and Pricing for TCP." IEEE JSAC Special Issue on Advances in the Fundamentals of Networking, Sep 1995, <URL:http://tiny-tera.stanford.edu/~nickm/papers.html>

[Fankhauser98] George Fankhauser, Burkhard Stiller, Christoph Vögtli and Bernhard Plattner (ETH Zürich), "Reservation-based Charging in an Integrated Services Network", INFORMS Telecommunications Conference, FL, 8-11 Mar 1998, <URL:ftp://ftp.tik.ee.ethz.ch/pub/people/stiller/paper/informs98.ps.gz>

[Floyd94] Sally Floyd (LBNL), "TCP and Explicit Congestion Notification", ACM Computer Communication Review, V. 24 N. 5, Oct 1994, p. 10-23. [This issue of CCR incorrectly has "1995" on the cover instead of "1994".] <URL:http://www-nrg.ee.lbl.gov/nrg-papers.html>

[Floyd95] Sally Floyd and Van Jacobsen, "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365-386, Aug 1995, <URL:http://www-nrg.ee.lbl.gov/floyd/cbq.html>

[Herzog95] Shai Herzog (IBM), Scott Shenker (Xerox PARC), Deborah Estrin (USC/ISI), "Sharing the cost of Multicast Trees: An Axiomatic Analysis", in Proceedings of ACM/SIGCOMM '95, Cambridge, MA, Aug. 1995, <URL:http://www.research.ibm.com/people/h/herzog/sigton.html>

[IEEE802.1p] IEEE, "Draft Standard for Traffic Class and Dynamic Multicast Filtering Services in Bridged Local Area Networks (Draft Supplement to 802.1D)", IEEE802.1p, 1997

[ITU_D.150] ITU-T, "New system for accounting in international telephony", ITU-T Rec. D.150, Series D: General tariff principles - Charging and accounting in the international telephone service; Oct 1996, <URL:http://www.itu.ch/intset/itu-t/d150/d150.htm>

[Kelly97] Frank P Kelly (Cambridge Uni.), "Charging and Accounting for Bursty Connections", in "Internet Economics" (Editors Lee W. McKnight and Joseph P. Bailey) MIT Press, 1997. 253-278. <URL:http://www.statslab.cam.ac.uk/~frank/charge.html>

[Kelly98] Frank P Kelly, Aman K. Maulloo and David KH Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability", Journal of the Operational Research Society, 49,1998, <URL:http://www.statslab.cam.ac.uk/~frank/rate.html>

[Linington98] Peter Linington, Zoran Milosevic, Kerry Raymond (Uni of Kent at Canterbury, UK), "Policies in Communities: Extending the ODP Enterprise Viewpoint", Proc EDOC'98, San Diego, Nov 1998,

[May97] Martin May, Jean-Chrysostome Bolot, Christophe Diot and Alain Jean-Marie (INRIA), "1-Bit Schemes for Service Discrimination in the Internet: Analysis and Evaluation", Aug 1997, <URL:http://www.inria.fr/rodeo/personnel/mmay/papers/rr_1bit.ps>

[Nichols97] Kathleen Nichols (Bay), Van Jacobson (LBL), Lixia Zhang (UCLA), "A Two-bit Differentiated Services Architecture for the Internet", work in progress: Internet Draft, Nov 1997, <URL:http://diffserv.lcs.mit.edu/Drafts/draft-nichols-diff-svc-arch-00.pdf>

[Odlyzko97] Andrew Odlyzko (AT&T Research), "A modest proposal for preventing Internet congestion", Sep 1997, <URL:http://www.research.att.com/~amo/doc/recent.html>

[Odlyzko98] Andrew Odlyzko (AT&T Research), "The economics of the Internet: Utility, utilization, pricing, and Quality of Service", Jul 1998, <URL:http://www.acm.org/sigcomm/sigcomm98/tp/abs_01.html>

[RFC1633] R. Braden, D.Clark, S.Shenker, "Integrated Services in the Internet architecture: an overview", IETF RFC 1633, Jun 1994. <URL:http://www.isi.edu/div7/rsvp/pub.html>

[RFC1889] H. Schulzrinne (GMD Fokus), S. Casner (Precept), R. Frederick (Xerox PARC), V. Jacobson (LBNL), "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 1889, Jan 1996, <URL:rfc1889.txt>

[RFC2001] W. Stevens (NOAO), "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", IETF RFC2001, Jan 1997 <URL:rfc2001.txt>

[RF 063] N. Brownlee, C. Mills, G. Ruth, "Traffic Flow Measurement: Architecture" IETF RFC 2063, Jan 1997 <URL:rfc2063.txt>

[RFC2208] F. Baker, B. Braden, S. Bradner, A. Mankin, M. O"Dell, A. Romanow, A. Weinrib, L. Zhang, "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment", IETF RFC 2208, Jan 1997, <URL:rfc2208.txt>

[RFC2327] Mark Handley, Van Jacobsen, "SDP: Session Description Protocol", IETF RFC 2327, Mar 1998, <URL:rfc2327.txt>

[RFC2475] S. Blake (Torrent), D. Black (EMC), M. Carlson (Sun), E. Davies (Nortel), Z. Wang (Bell Labs Lucent), W. Weiss (Lucent), "An Architecture for Differentiated Services", IETF RFC 2475, Dec 1998 <URL:rfc2475.txt>

[Ruth97] Gregory R. Ruth (GTE), "Usage Accounting for the Internet", In Proc. INET'97 Kuala Lumpur, <URL:http://www.isoc.org/isoc/whatis/conferences/inet/97/proceedings/F1/F1_1.HTM>

[Shenker95] Scott Shenker (Xerox PARC), "Fundamental Design Issues for the Future Internet", IEEE Journal on Selected Areas in Communications, 1995. URL: <http://www.spp.umich.edu/spp/courses/744/docs/FundamentalDesign-shenker.pdf>

[Shenker96] Scott Shenker (Xerox PARC), David Clark (MIT), Deborah Estrin (USC/ISI) and Shai Herzog (USC/ISI), 'Pricing in Computer Networks: Reshaping the research agenda', SIGCOMM Computer Communication Review Volume 26, Number 2, Apr 1996, <URL: http://www.statslab.cam.ac.uk/~frank/PRICE/scott.ps>

[Stiller98] Burkhard Stiller, George Fankhauser, Bernhard Plattner and Natalie Weiler, (ETH Zürich), "Charging and Accounting for Integrated Internet Services - State of the Art, Problems, and, Trends", INET'98, Jul 1998, <URL:ftp://ftp.tik.ee.ethz.ch/pub/people/stiller/paper/inet98.ps.gz>

[TasselBri97] Jérôme Tassel, Bob Briscoe, Alan Smith, (BT), "An End to End Price-Based QoS Control Component Using Reflective Java", in Lecture Notes in Computer Science from the 4th COST237 workshop, pub. Springer-Verlag, Dec 1997, <URL:http://www.labs.bt.com/people/briscorj/papers.html#QoteS>

[xxxa99] XXX {anonymised for double blind reviewing}, "The Direction of Value Flow in Connectionless Networks", forthcoming. {Attached}

[Zhang93] Lixia Zhang (Xerox PARC), Stephen Deering (Xerox PARC), Deborah Estrin(USC/ISI), Scott Shenker (Xerox PARC) and Daniel Zappala (USC/CSD), "RSVP: A New Resource ReSerVation Protocol", IEEE Network. Sep 1993. <URL:http://www.isi.edu/div7/rsvp/pub.html>

CLAIMS

1. A method of operating a communications network comprising:
   a) measuring at each of a plurality of customer terminals usage by the respective customer terminal of network resources; and
   b) subsequently calculating a network usage charge from the measurement data generated by step (a).

2. A method of operating a federated data communications network characterised by measuring at each of a plurality of customer terminals connected to the said network usage by the respective customer terminal of network resources.

3. A method according to claim 2, further comprising subsequently calculating a network usage charge from measurement data generated by the step of measuring.

4. A method according to any one of the preceding claims, further comprising step of aggregating measurement data produced by a series of measurements at respective customer terminal.

5. A method according to any one of the preceding claims, further comprising storing the measurement data.

6. A method according to claim 5, including storing with the measurement data data identifying a tariff applicable to the said measurement data.

7. A method according to any one of the preceding claims including communicating data generated by step (a) to a network accounting object controlled by a network operator.

8. A method according to claim 7, including communicating to the network accounting object a usage charge calculated from the measurement data.

9. A method according to any one of the preceding claims, including communicating measurement data to a system remote from the customer terminal.

10. A method according to any one of the preceding claims, including a step carried out by the network operator of sampling part only of the traffic communicated between a customer terminal and the network and for the sampled traffic comparing the network usage with data communicated from the customer terminal to the network accounting object and thereby detecting any discrepancy.

11. A method according to any one of the preceding claims in which a network accounting object is configurable to receive data either from a measurement object controlled by the network operator or from a customer terminal.

12. A method according to claim 11, in which a customer accounting object associated with the customer terminal is configurable to direct data to the network accounting object.

13. A method according to claim 11 or 12, including switching the network accounting object from a first configuration in which data is received from the said measurement object and another configuration in which data is received from the customer terminal in response to a control signal received at the network accounting object.

14. A method according to any one of the preceding claims further comprising communicating a tariff to each of the customer terminals, and calculating at each of the terminals from the tariff and from the accounting data the network usage .charge.

15. A method according to any one of the preceding claims in which the communications network is a federated data network comprising a plurality of network domains.

16. A method according to claim 15 including

communicating traffic between a customer terminal and a first network domain connected to the customer terminal,

further communicating the said traffic between the first network domain and a second network domain connected to the first network domain;

communicating network usage data from the customer terminal to a first network accounting object in the first domain;

communicating accounting data between the first network accounting object and a second network accounting object in the second domain.

17. A method according to claim 16, including determining from a current routing table in the first network domain the identity of a second domain, which second domain is communicating data with the customer terminal via the first network domain, and communicating network usage data for the customer terminal to the second domain identified by the current routing table.

18. A method according to any one of the preceding claims in which the step of measuring includes counting the number of packets communicated between the customer terminal and the communications network.

19. A method according to claim 18, including measuring both packets received by the customer terminal and packets sent by the customer terminal.

20. A method according to any one of the preceding claims, in which a payment for network usage is made to a third-party clearer.

21. A communications network arranged to operate by a method according to anyone of the preceding claims.

22. A customer terminal arranged to operate by a method according to any one of the preceding claims.

23. A customer terminal including a data interface arranged to be connected to a federated data network, characterised by a network usage meter arranged to measure the usage by the customer terminal of network resources.

24. A customer terminal according to claim 23, in which the usage meter includes means for counting the number of packets communicated between the customer terminal and the network via the data interface.

25. A customer terminal according to any of claims 22 to 24, including an accounting interface arranged to communicate measurement data to a network accounting object.

26. A method of operating a network comprising a plurality of network domains, including calculating a charge for use by a respective customer of network resources, and making payment in settlement of the said charge to a third party clearer.

27. A method according to any one of claims 1 to 20, including automatically varying a tariff for network usage in dependence on loading of the network, and calculating a charge for network usage by applying the tariff to the measurement data.

28. A method according to anyone of the preceding claims, including transmitting packets on the network at a plurality of different service levels.

29. A method according to claim 28, including passing the said packets through a packet router, and in the packet router determing a classification of packets, and scheduling packets differently depending on the packet classification.

30. A method according to claim 29, in which a step of policing the classification of packets to determine the eligibility of a packet for a respective service class is carried out at a location remote from the router.

31. A method according to claim 30, in which the step of policing is carried out at a customer terminal.

32. A router for use in a packet network providing a plurality of different service levels, the router comprising

means for determining the class of a packet received at a router,

means for scheduling a packet depending on its class,

in which the router is arranged to schedule packets depending on their class without policing the eligibility of a packet for a requested class of service.

33. A method of operating a packet network providing a plurality of different service levels, the method including including passing the said packets through a packet router, and in the packet router determining a classification of packets, scheduling packets differently depending on the packet classification and, at a location remote from the router, policing the service levels of packets to determine the eligibility of a packet for a respective service class .

34. A method of operating a federated communications network comprising a plurality of network domains, the method including determining a price for a data flow from one domain into an adjacent domain by:

a) announcing, by the one domain, a price for receiving the said data flow and for passing the said data flow from the one domain to the adjacent domain;

b) announcing, by the adjacent domain, a price for receiving the said data flow from the one domain and transmitting it onwards;

c) calculating an edge price for the data flow from the difference between the price announced in step (a) and the price announced in step(b).

35. A method of pricing data flows in a federated data network substantially as described in the attached paper "The Direction of Value Flow in Connectionless Networks".

36. A method of operating a federated communications network, in which a charge is applied to a data flow between end users, and a clearing payment is made by

one or more end users to a clearing entity via a route independent of the path of the data flow.

37. A method according to claim 10, or any one of the preceding claims when dependent on claim 10, including penalising a customer when a discrepancy is detected.

ABSTRACT

In a communications network, which may be a federated data network such as the Internet, use of network resources is measured locally at customer terminals, for example by counting the number of packets sent and received. The resulting data may be aggregated and sent to a network accounting object. Accounting data may subsequently be passed between network subdomains .

Figure 1

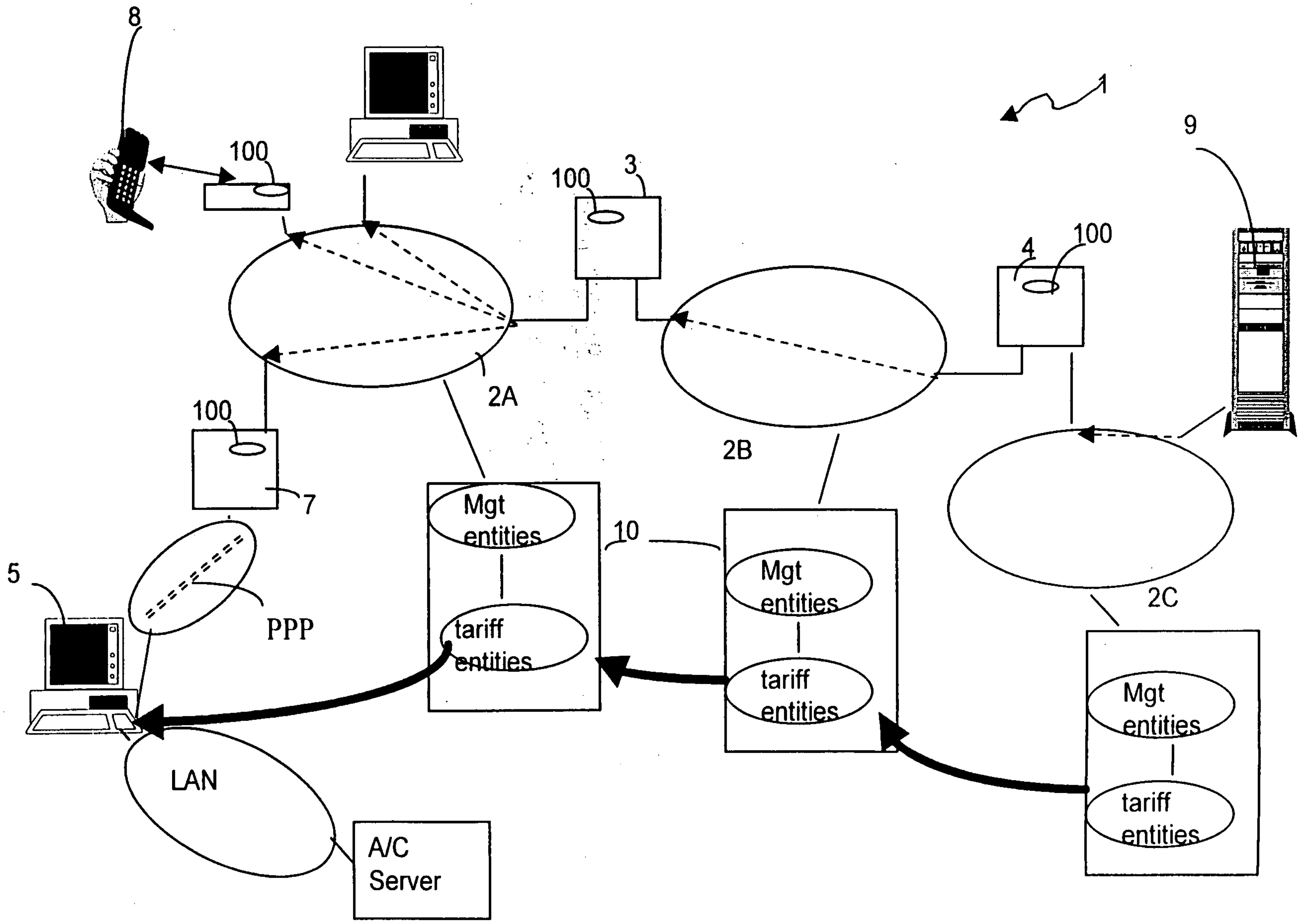

8
100
100
3
1
9
4
100
2A
100
2B
7
Mgt entities
10
Mgt entities
2C
5
PPP
tariff entities
tariff entities
Mgt entities
LAN
tariff entities
A/C Server

Figure 2a

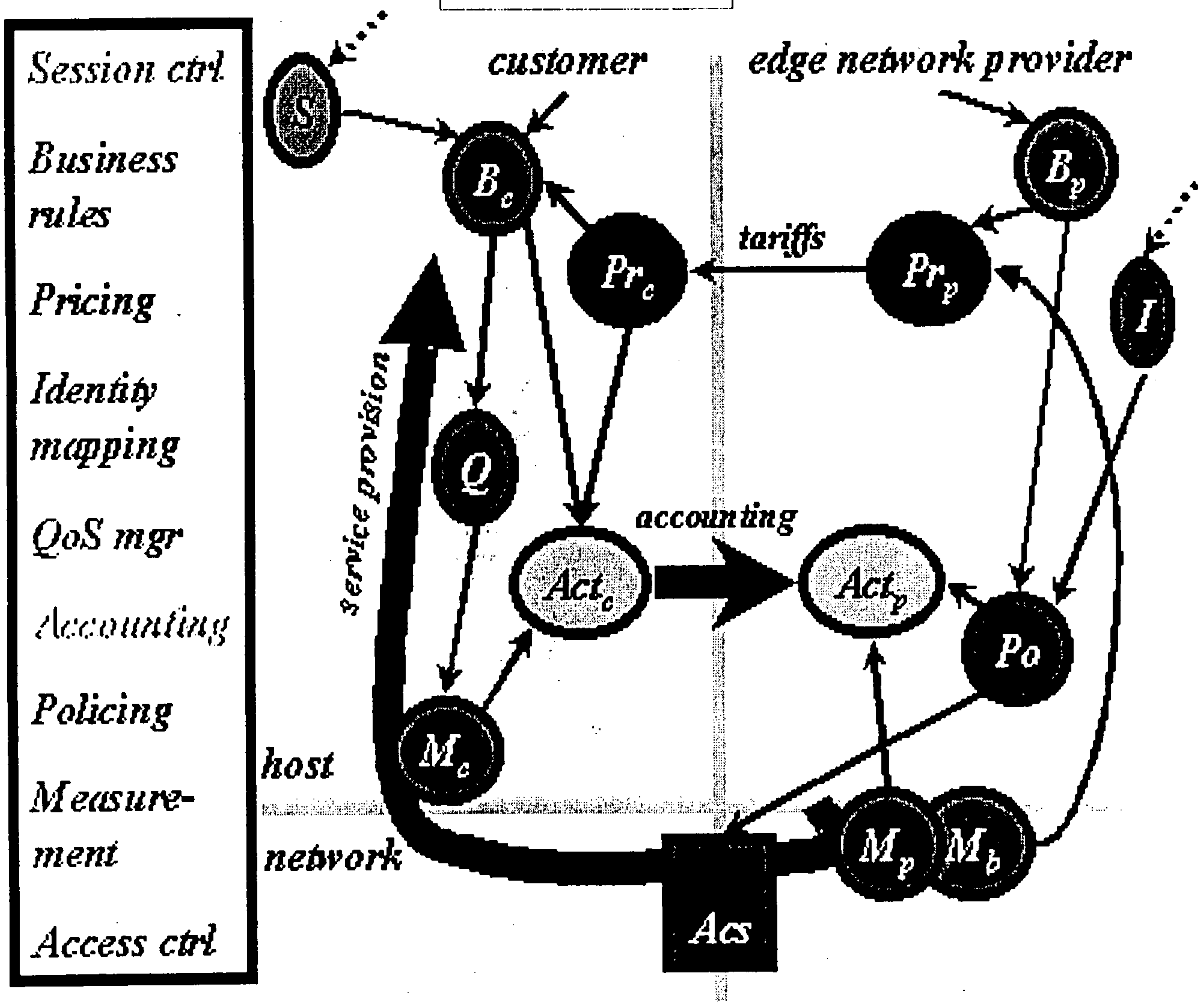

Session ctrl
Business rules
Pricing
Identity mapping
QoS mgr
Accounting
Policing
Measurement
Access ctrl
customer
edge network provider
S
B_c
Pr_c
B_p
Pr_p
I
tariffs
Q
Act_c
accounting
Act_p
Po
service provision
M_c
host
network
Acs
M_p
M_b

Figure 2b

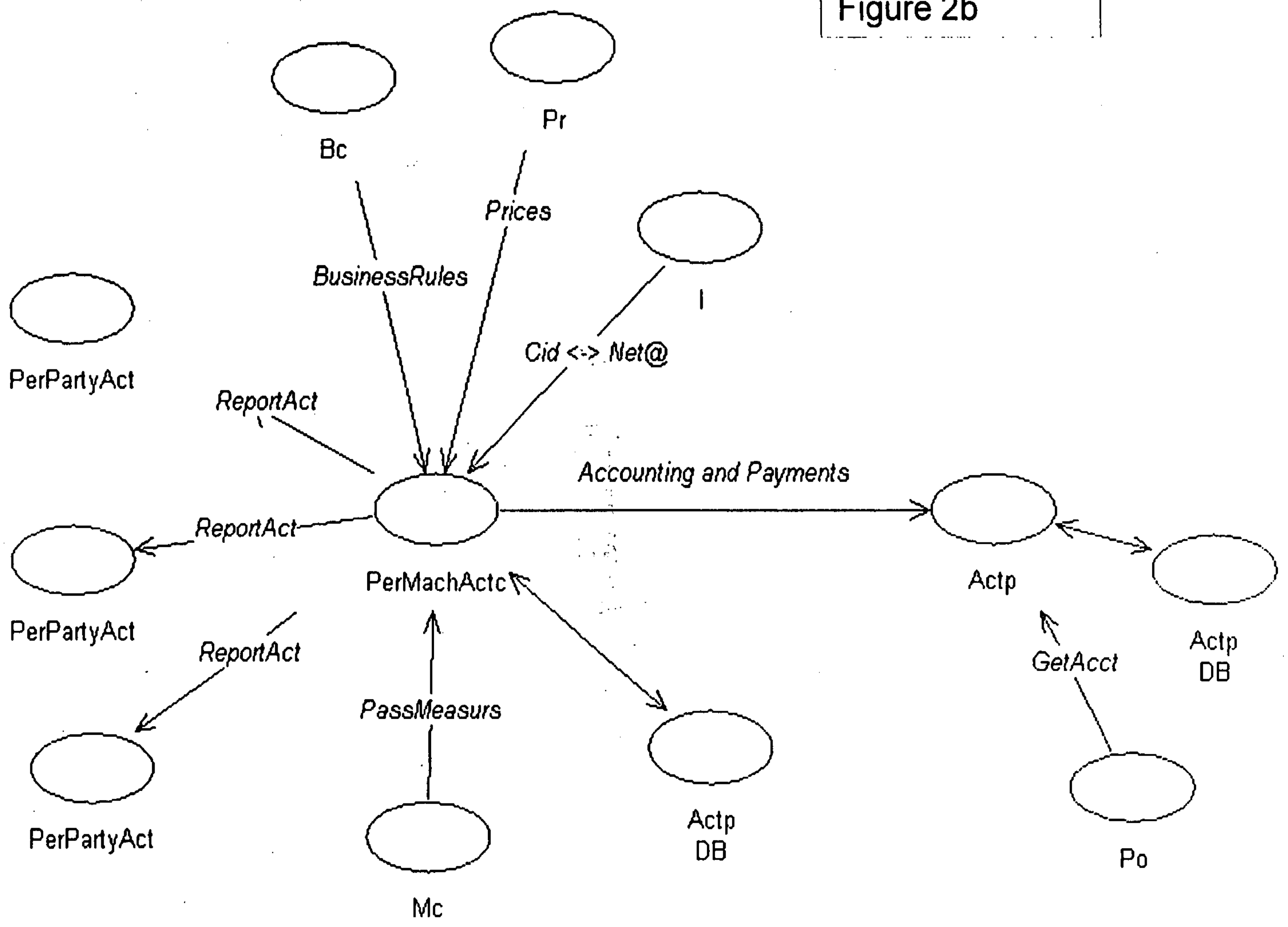

Bc
Pr
Prices
BusinessRules
I
Cid <-> Net@
PerPartyAct
ReportAct
Accounting and Payments
ReportAct
PerPartyAct
PerMachActc
Actp
Actp
DB
ReportAct
GetAcct
PassMeasurs
PerPartyAct
Actp
DB
Po
Mc

Figure 3a

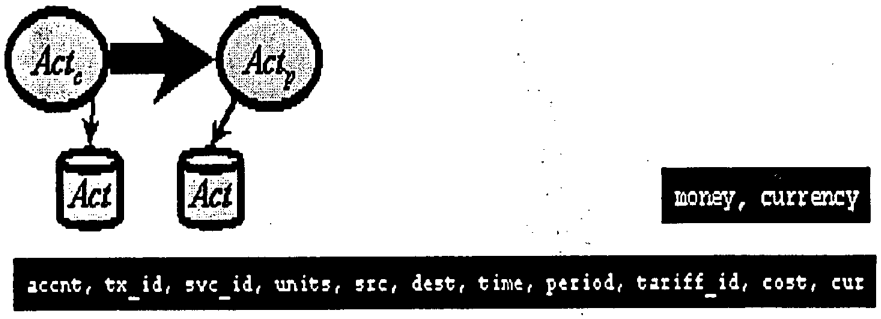

Act_c
Act_p
Act
Act
money, currency
accnt, tx_id, svc_id, units, src, dest, time, period, tariff_id, cost, cur

Figure 3b

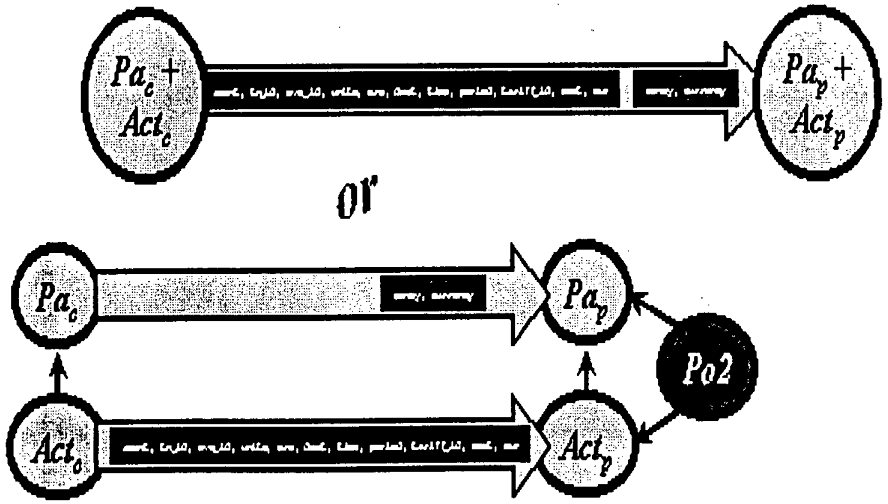
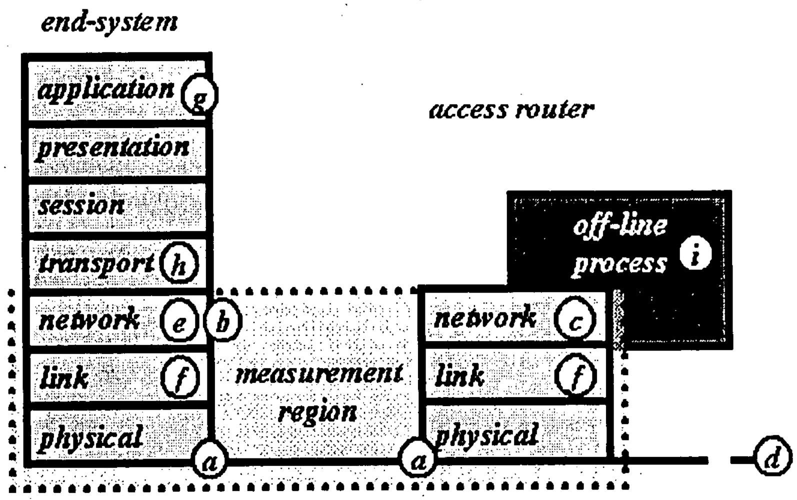


Figure 4



end-system
application g
presentation
session
transport h
network e
b
link f
physical
a
measurement region
a
access router
off-line process i
network c
link f
physical
d

Figure 5a

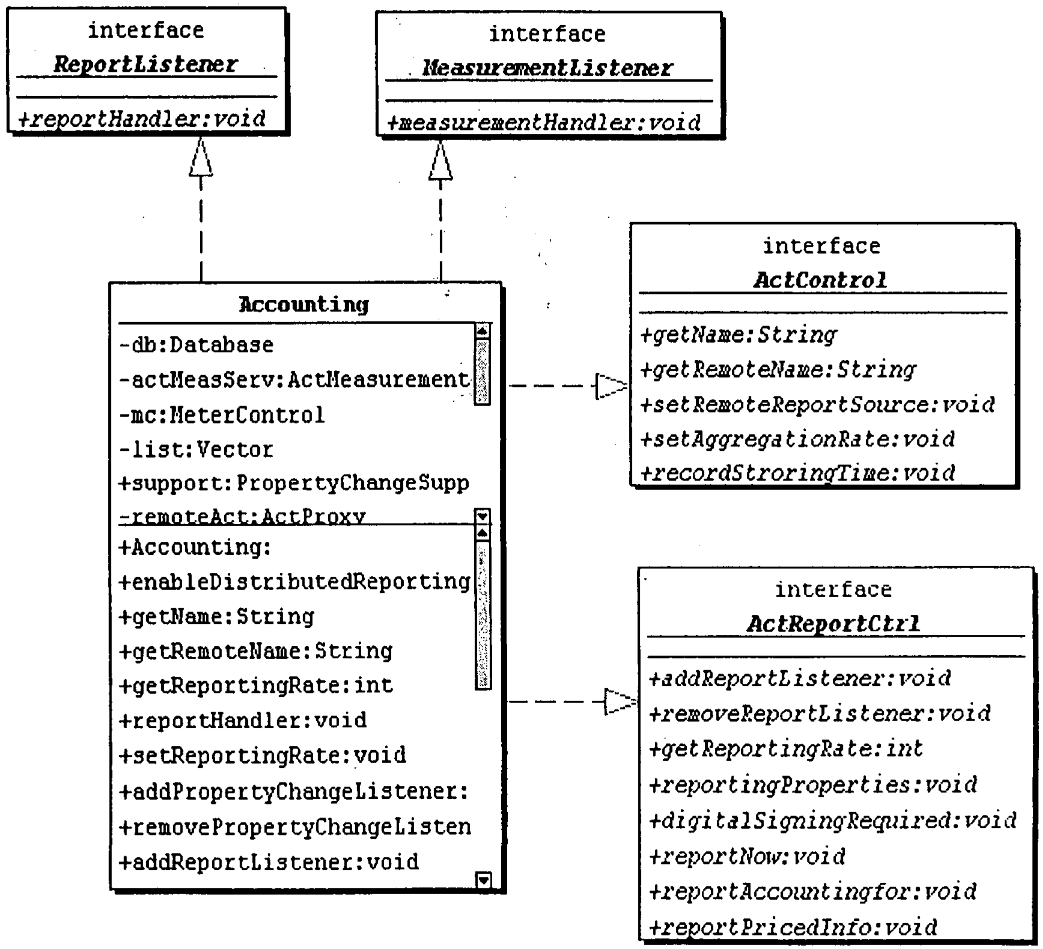

interface
ReportListener
+reportHandler:void
interface
MeasurementListener
+measurementHandler:void
Accounting
-db:Database
-actMeasServ:ActMeasurement
-mc:MeterControl
-list:Vector
+support:PropertyChangeSupp
-remoteAct:ActProxy
+Accounting:
+enableDistributedReporting
+getName:String
+getRemoteName:String
+getReportingRate:int
+reportHandler:void
+setReportingRate:void
+addPropertyChangeListener:
+removePropertyChangeListen
+addReportListener:void
interface
ActControl
+getName:String
+getRemoteName:String
+setRemoteReportSource:void
+setAggregationRate:void
+recordStroringTime:void
interface
ActReportCtrl
+addReportListener:void
+removeReportListener:void
+getReportingRate:int
+reportingProperties:void
+digitalSigningRequired:void
+reportNow:void
+reportAccountingfor:void
+reportPricedInfo:void

Figure 5b

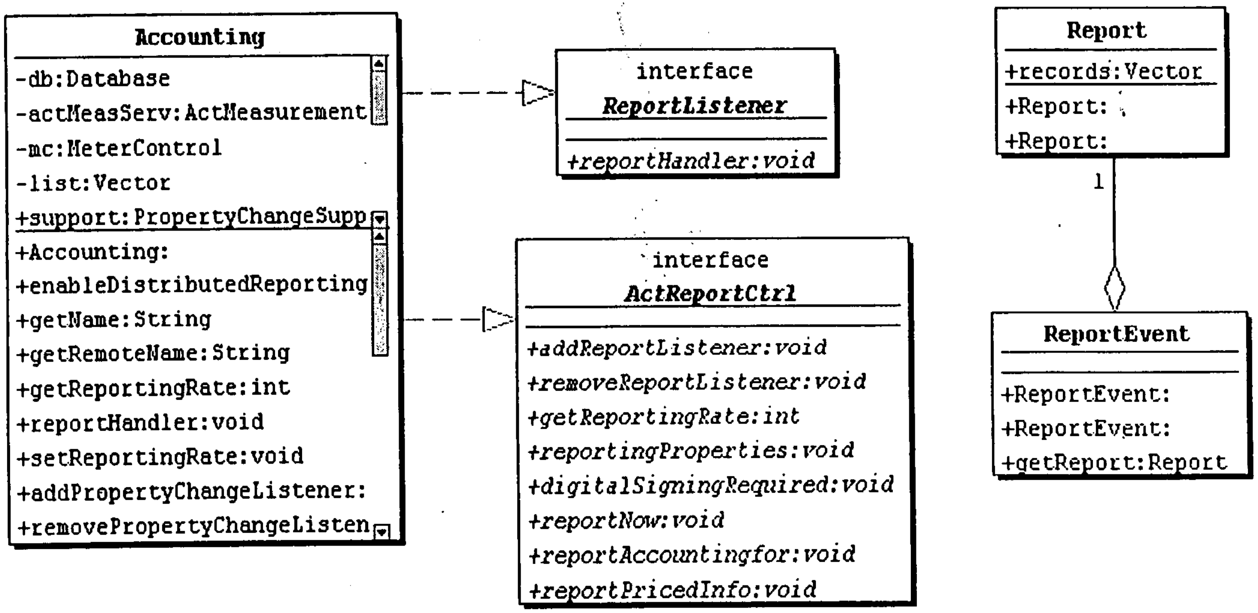

Accounting
-db:Database
-actMeasServ:ActMeasurement
-mc:MeterControl
-list:Vector
+support:PropertyChangeSupp
+Accounting:
+enableDistributedReporting
+getName:String
+getRemoteName:String
+getReportingRate:int
+reportHandler:void
+setReportingRate:void
+addPropertyChangeListener:
+removePropertyChangeListen
interface
ReportListener
+reportHandler:void
interface
ActReportCtrl
+addReportListener:void
+removeReportListener:void
+getReportingRate:int
+reportingProperties:void
+digitalSigningRequired:void
+reportNow:void
+reportAccountingfor:void
+reportPricedInfo:void
Report
+records:Vector
+Report:
+Report:
1
ReportEvent
+ReportEvent:
+ReportEvent:
+getReport:Report

Figure 5c

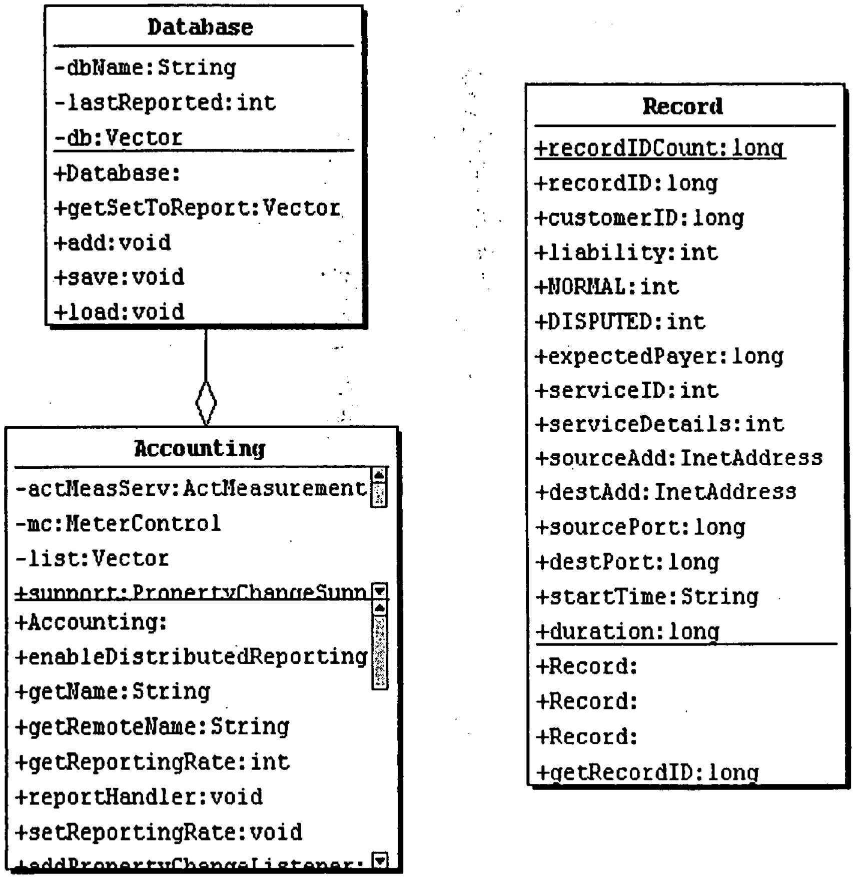

Database
-dbName:String
-lastReported:int
-db:Vector
+Database:
+getSetToReport:Vector
+add:void
+save:void
+load:void
Accounting
-actMeasServ:ActMeasurement
-mc:MeterControl
-list:Vector
+support:PropertyChangeSupp
+Accounting:
+enableDistributedReporting
+getName:String
+getRemoteName:String
+getReportingRate:int
+reportHandler:void
+setReportingRate:void
+addPropertyChangeListener:
Record
+recordIDCount:long
+recordID:long
+customerID:long
+liability:int
+NORMAL:int
+DISPUTED:int
+expectedPayer:long
+serviceID:int
+serviceDetails:int
+sourceAdd:InetAddress
+destAdd:InetAddress
+sourcePort:long
+destPort:long
+startTime:String
+duration:long
+Record:
+Record:
+Record:
+getRecordID:long

Figure 5d

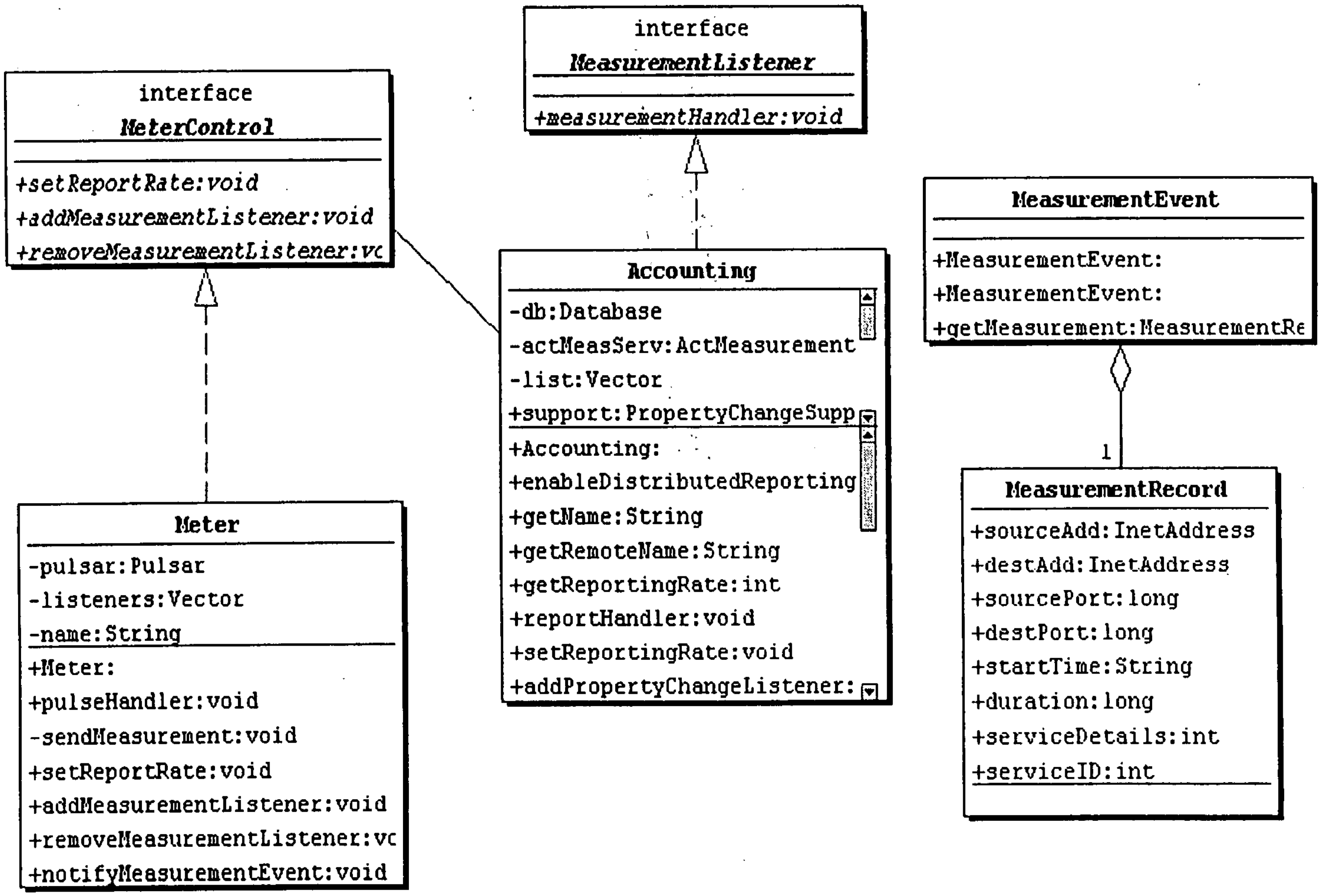

interface
MeterControl
+setReportRate:void
+addMeasurementListener:void
+removeMeasurementListener:vc
interface
MeasurementListener
+measurementHandler:void
MeasurementEvent
+MeasurementEvent:
+MeasurementEvent:
+getMeasurement:MeasurementRe
Accounting
-db:Database
-actMeasServ:ActMeasurement
-list:Vector
+support:PropertyChangeSupp
+Accounting:
+enableDistributedReporting
+getName:String
+getRemoteName:String
+getReportingRate:int
+reportHandler:void
+setReportingRate:void
+addPropertyChangeListener:
1
MeasurementRecord
+sourceAdd:InetAddress
+destAdd:InetAddress
+sourcePort:long
+destPort:long
+startTime:String
+duration:long
+serviceDetails:int
+serviceID:int
Meter
-pulsar:Pulsar
-listeners:Vector
-name:String
+Meter:
+pulseHandler:void
-sendMeasurement:void
+setReportRate:void
+addMeasurementListener:void
+removeMeasurementListener:vc
+notifyMeasurementEvent:void

Figure 5e

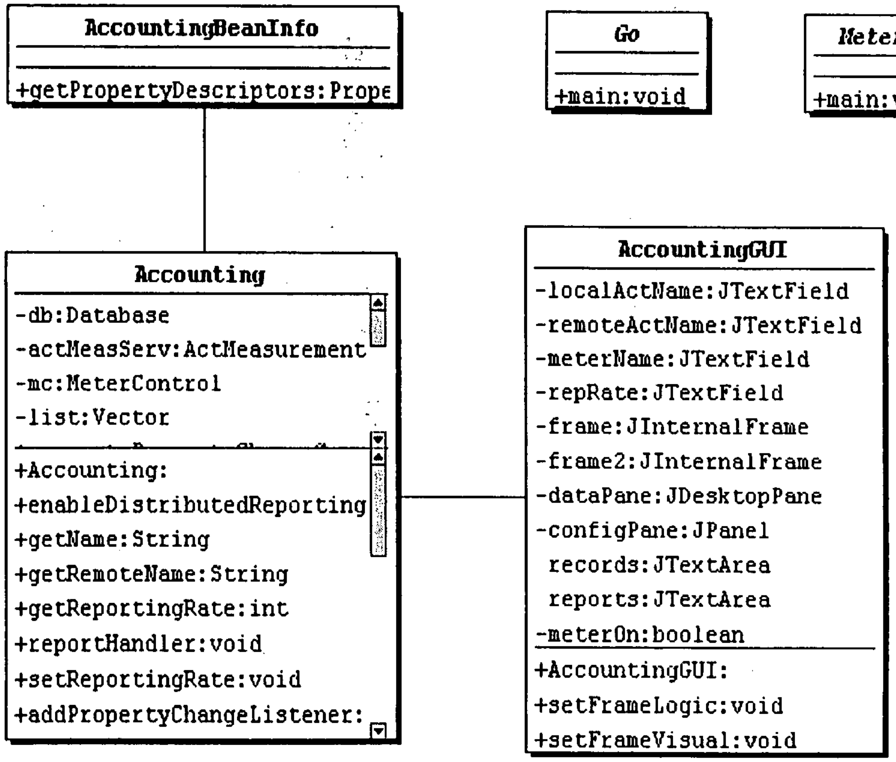
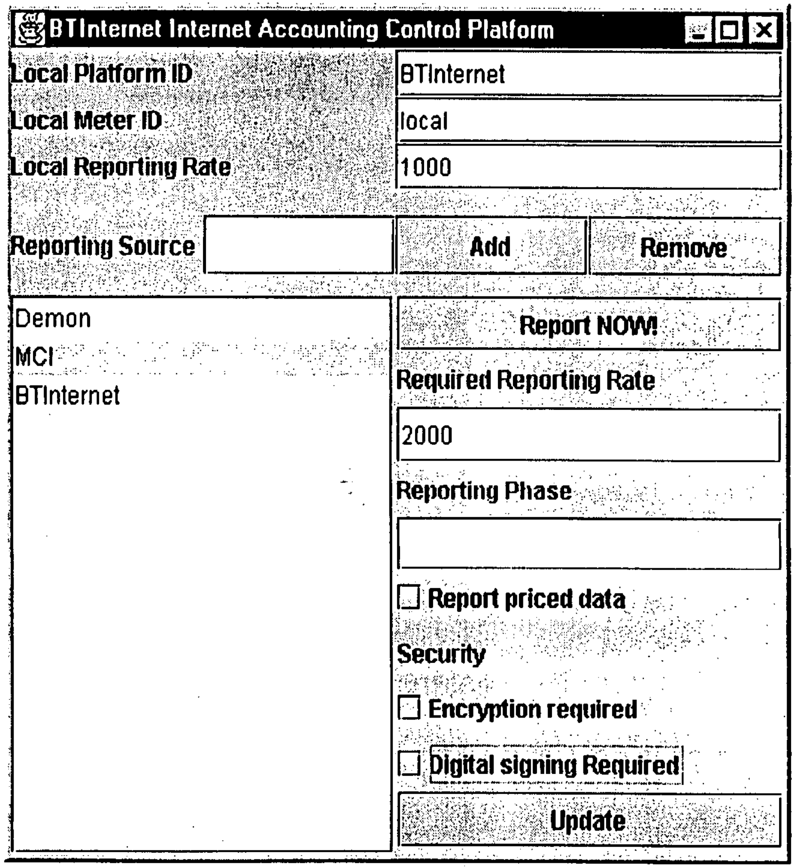

AccountingBeanInfo
+getPropertyDescriptors:Prope
Go
+main:void
MeterGo
+main:void
Accounting
-db:Database
-actMeasServ:ActMeasurement
-mc:MeterControl
-list:Vector
+Accounting:
+enableDistributedReporting
+getName:String
+getRemoteName:String
+getReportingRate:int
+reportHandler:void
+setReportingRate:void
+addPropertyChangeListener:
AccountingGUI
-localActName:JTextField
-remoteActName:JTextField
-meterName:JTextField
-repRate:JTextField
-frame:JInternalFrame
-frame2:JInternalFrame
-dataPane:JDesktopPane
-configPane:JPanel
records:JTextArea
reports:JTextArea
-meterOn:boolean
+AccountingGUI:
+setFrameLogic:void
+setFrameVisual:void

Figure 6



BTInternet Internet Accounting Control Platform

Local Platform ID: BTInternet

Local Meter ID: local

Local Reporting Rate: 1000

Reporting Source:

Demon
MCI
BTInternet

Required Reporting Rate: 2000

Reporting Phase:

□ Report priced data

Security

□ Encryption required

□ Digital signing Required

Figure 7

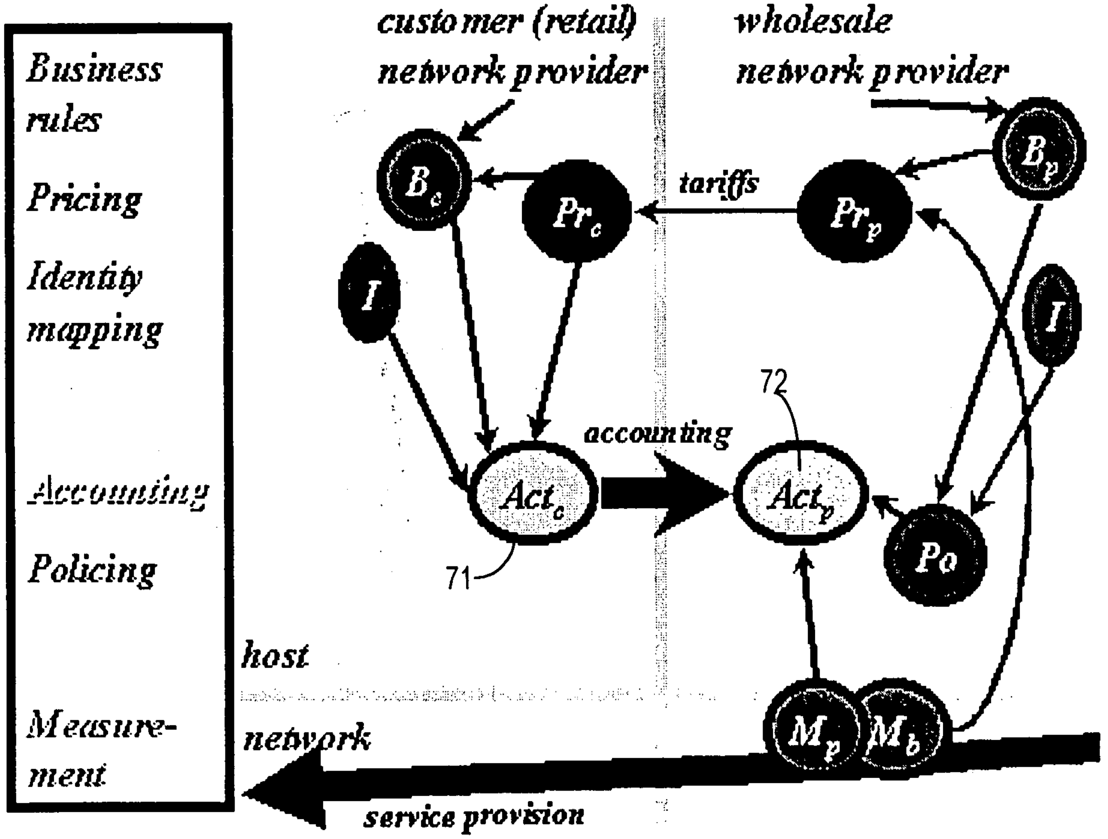

Business rules
Pricing
Identity mapping
Accounting
Policing
Measure-ment
customer (retail) network provider
wholesale network provider
B_c
Pr_c
tariffs
Pr_p
B_p
I
I
72
accounting
Act_c
Act_p
Po
71
host
network
M_p
M_b
service provision

Figure 8

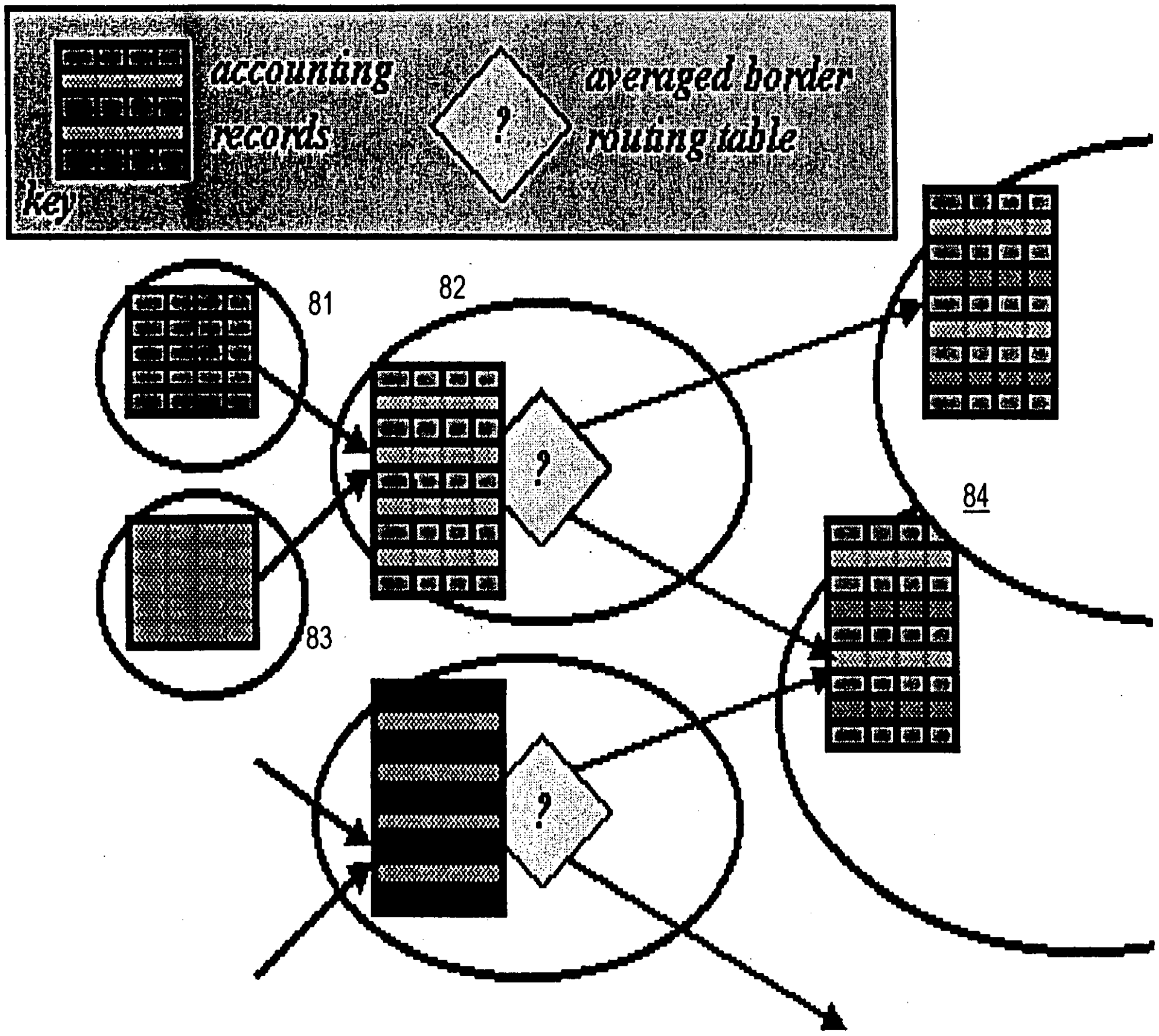

key
accounting records
?
averaged border routing table
81
82
83
84
?
?

Figure 9

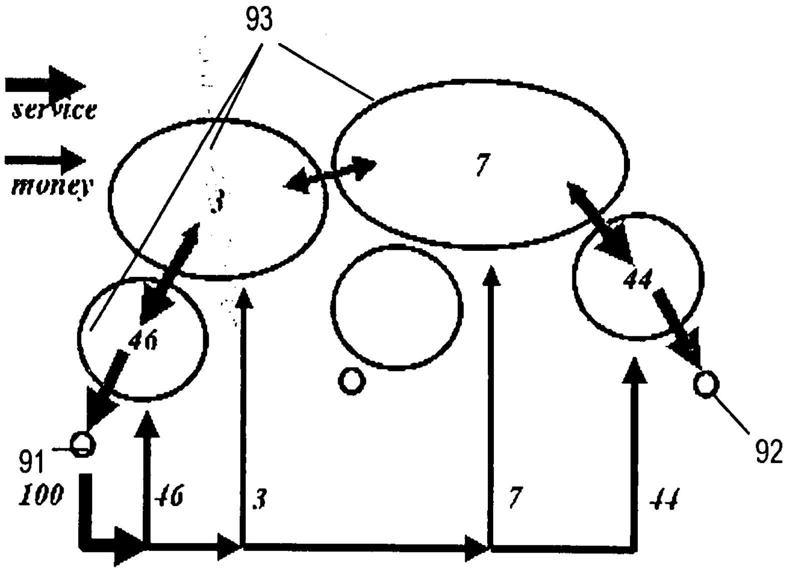
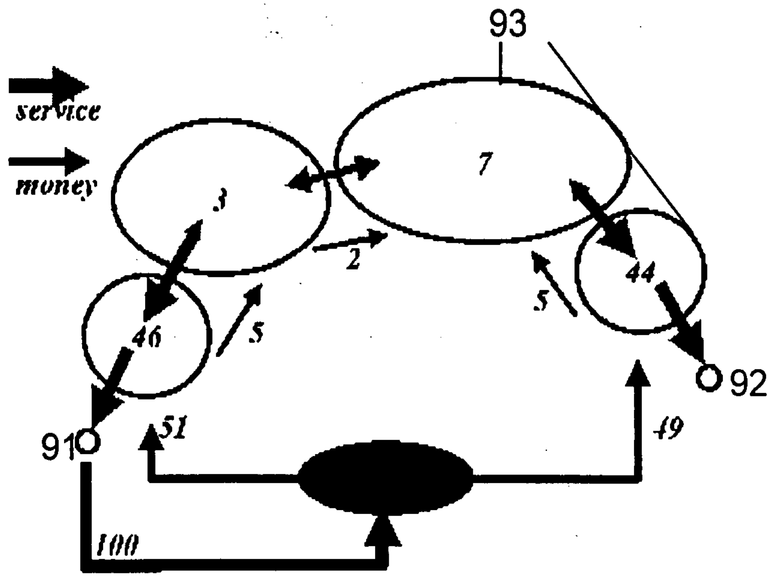


Figure 10



PCT/GB99/01772
4/6/00
BT Group